Agile Service Design Framework™

by John Parker

Contact Us:
210.399.4240
info@EnfocusSolutions.com

January 2015
www.EnfocusSolutions.com

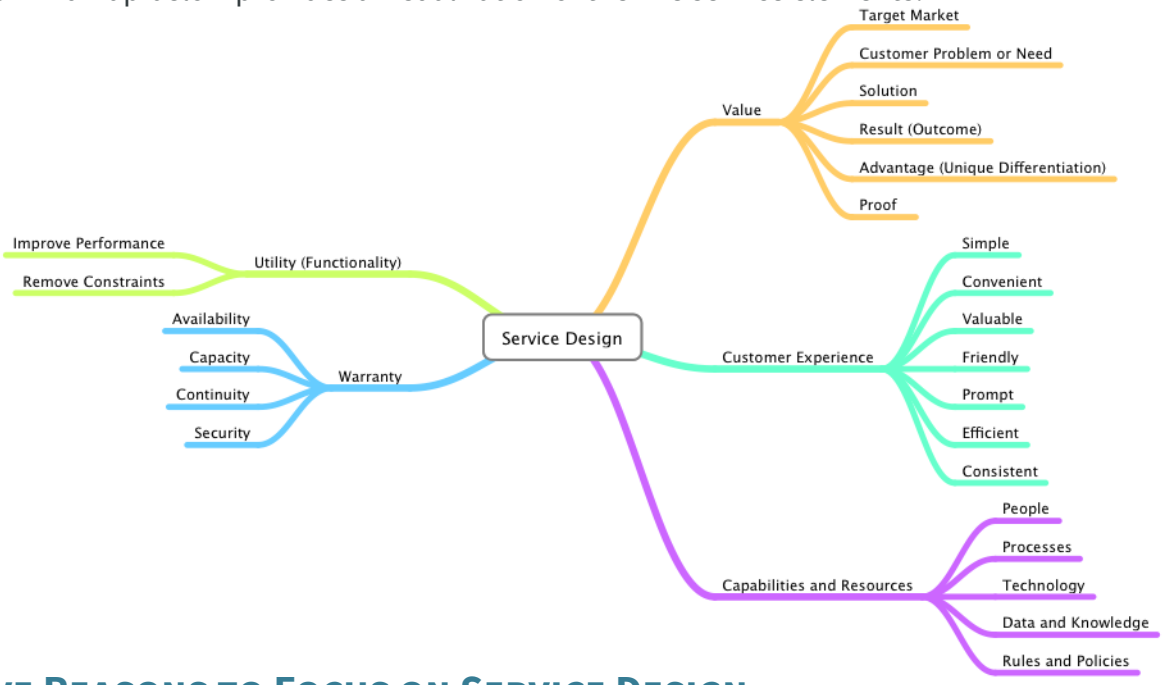
Table of Contents

What is Service Design?

Service Design is a method for designing experiences that reach people through many different touchpoints happening over time. Service Design, in the context of IT services, involves defining the architecture, processes, policies, and documentation, as well as ensuring current and future agreed to business requirements are met. The complete definition of a service consists of five key elements:

SERVICE ELEMENTS	DEFINED BY	DESCRIPTION
 **Utility**	• Service Features • Functional Requirements	Utility is about the functionality the service provides, and is achieved by increasing performance (i.e., being able to do more in less time) or eliminating constraints (i.e., restrictions or inabilities to execute a specific task).
 **Warranty**	• Service Level Requirements • (Nonfunctional Requirements)	Warranty from an IT service management perspective addresses four key questions: • Is the service available enough? • Is there enough capacity? • Is it secure enough? • Is it continual enough?
 **Value**	• Value Propositions	Business value is achieved when people use the service to perform meaningful work, resulting in better business outcomes.
 **Customer Experience**	• Touchpoints	Customer experience is the sum of all experiences a customer has with a supplier of goods and/or services over the duration of the relationship. This can include awareness, discovery, attraction, interaction, purchase, use, cultivation, and advocacy.
 **Capabilities and Resources**	• Components • Related Services	To deliver a quality service requires having the right capabilities and resources. According to ITIL, capabilities refer to the abilities of an organization, person, process, application, configuration item, or IT service to carry out an activity; and, resources refer to the IT infrastructure, people, money, or anything else that might help to deliver an IT service.

The mindmap below provides a visualization of the five service elements.



Five Reasons to Focus on Service Design

As more and more organizations adopt Service Design best practices, it becomes increasingly apparent how valuable the discipline is. Unfortunately, many organizations are still giving their attention to product design rather than the overall service. Here's five reasons to focus on Service Design if you're not already:

1. The service industry accounts for around 75% of the Western economy and represents 80% of the US economy. Services are rarely designed with the same care and attention to detail as products. However, better designed services lead to:
   - Increased sales from customers who are willing to pay more,
   - Greater customer loyalty and retention, and
   - Lower costs though more efficient business processes.
2. A recent survey titled "Finding the Performance Pay-off in Customer Experience" gathered the opinion of 644 business leaders. The survey showed that organizations with the strongest use of customer experience in daily decision making report the strongest operating results, with 67% reporting that they were able to meet or exceed revenue targets, and 65% having met or exceeded profit targets. These organizations use customer experience as a planned strategy for achieving performance.
3. Another study conducted by Peer Insight of Fortune 500 companies reported that companies focused on customer experience design consistently outperformed the S&P 500 by a 10:1 margin.
4. What the experts have to say of its importance: "Mobile behaviors and user expectations for an engaging digital experience are changing. People now desire a more intuitive and meaningful interaction from both brands and their services. That's why we put them at the center of what we do." – *Kristina Dervojeda, et. al., Design for Innovation: Service design as a means to advance business models*

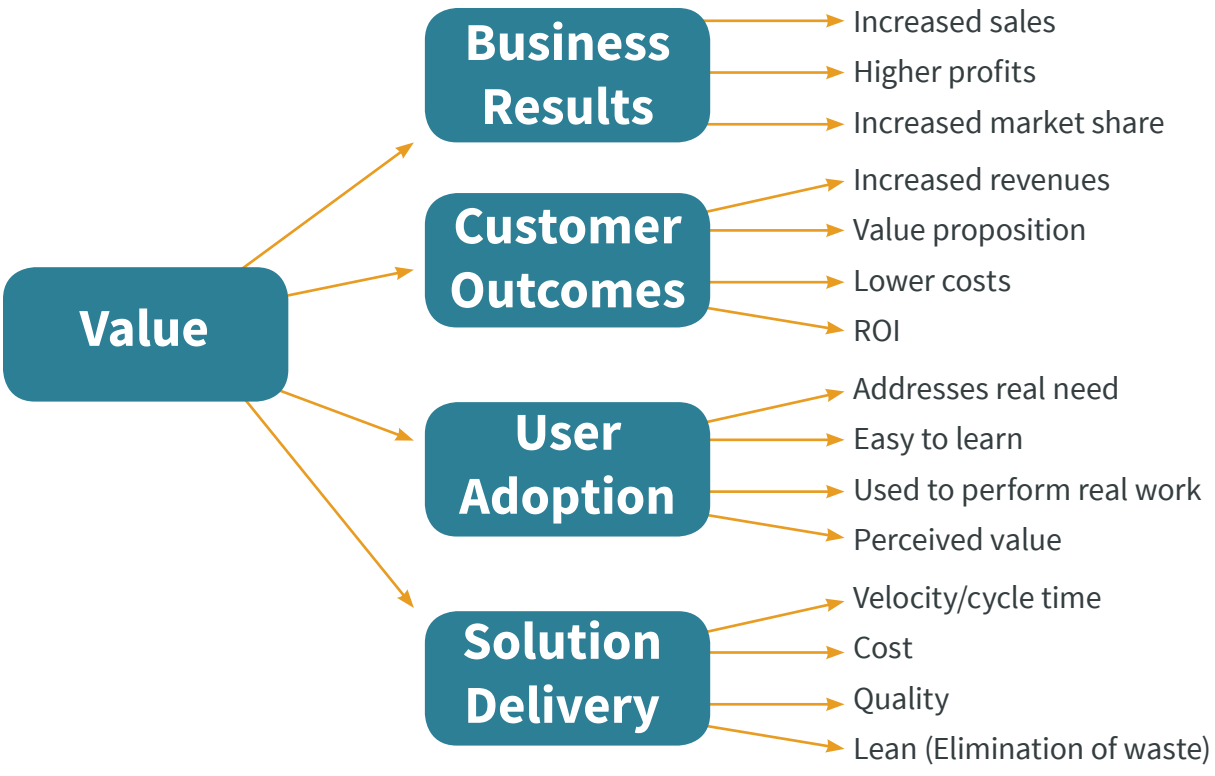
⑤ In a recent interview, managing director and chief business designer for Cultivar Consulting Limited, James Rock, discussed the relationship between Service Design and the customer: "Customers are… becoming more demanding so it is… very important that… service organizations develop highly responsive service recovery processes. In the rapidly growing world of social media, customers are becoming more vocal and very quick to complain about poor service to thousands of friends and followers on Facebook and Twitter. This 'word of mouth' effect is playing a bigger and bigger role in brand marketing campaigns. So I think it's only natural that organizations recognize they need to constantly improve and reinvent the way [services are] delivered to make sure they delight rather than disappoint customers."

The Business Value of Service Design

The overlying goal of a service is—simply—to provide business value. Business value is achieved when people use the service to perform meaningful work resulting in better business outcomes. Here are four principles to keep in mind when using services to deliver value to customers:

1. Software provides no value unless it is used.
2. Business value is achieved when people use software to perform meaningful work, resulting in better business outcomes.
3. Value must be continuously discovered and validated.
4. Value must be measured and managed.

We've discovered that there are four key aspects to delivering business value, shown in the diagram below. Value may come in many forms:



There are often obstacles to delivering business value. These items are common potential problems for Service Design teams that will, if encountered, impair the delivery of business value:

- An ineffective Inspect and Adapt process.
- Making the mistake of using a single backlog for epics, features, and user stories.
- Projects that are primarily technology-focused instead of business change-focused.
- Failure to define a clear vision.
- Failure to address key user needs in the solution, resulting in the business failing to adopt the solution.
- Poorly documenting the business problem, resulting in a flawed business case.
- Poorly developing the business case and establishing an incorrect or unrealistic expectation.
- Inaccurate, incomplete, or poorly defined requirements.
- Poorly executed delivery.
- A fundamentally flawed technical solution.
- Significant changes occurring in the business between project inception and completion.
- Benefits that are not properly managed or harvested.

Key Service Design Concept: Collaborative Service Architecture

A key concept in Service Design is the Service Architecture, or the people, processes, rules data, and IT services that make. A documented Service Architecture ensures we have a method for handling project risk and impact. We document the Service Architecture to align the business' strategic vision with its information technology, ensuring we provide valuable services to customers. The Service Architecture connects different business units to ensure synergistic communication and collaboration, creating a more seamless customer (or user) experience. A well-defined Service Architecture results in improvements to…

- Customer experience/service design efforts
- Business process improvement initiatives
- Application rationalization
- Opportunities for revenue generation or cost reduction
- Risk assessment and compliance

One of the most important benefits a well-managed architecture provides is to create employee engagement. A well-designed and executed Service Architecture does this in two ways. First, it creates meaning by connecting every role from the CEO to the janitor with the vision, mission, and goals of the organization, as well as the services it provides. Every person in the organization can clearly understand how their work contributes to the overall success of the organization. Second, it provides a set of guardrails making it safe and productive for employees to take independent action. When employees act autonomously they feel competent and confident. They are much more invested in the outcome when their own decisions guide what they do.

The table below describes the existing types of Service Architecture components that should be documented in a central repository.

ARCHITECTURE COMPONENT	DESCRIPTION
People	Stakeholders involved in the purchase, use, and delivery of the service, including: • Customers • Users • External Suppliers • Internal Suppliers • Service Delivery Team
Process	Business processes involved in the acquisition, delivery, and operations of the service, including: • Frontstage Processes • Backstage Processes • Technical Support Processes
Supporting Services	ITIL 2011 describes supporting services as "IT services that support or 'underpin' the customer-facing services." They are generally not listed in the Service Catalog as they not directly orderable by customers. They usually have an OLA or Underpinning Contract associated with them. Examples of supporting services include: • Server Management • Storage Management • Network Management • Security Management
Components	Defined in ISO/IEC 20000-1:2011 as a "Single unit of a service that when combined with other units will deliver a complete service." A component may consist of one or more configuration items. Examples include: • Hardware • Software • Tools • Applications • Documentation

Data and Knowledge	Data is becoming increasingly important in the delivery of services, especially with the advent of Big Data and Analytics services. These architectural components include: • Master Data • Service Knowledge • Customer Knowledge
Business Rules	Defining business rules is increasingly important for risk and compliance. For example, banks and insurance companies use business rules to manage compliance for financial services. Rules are often maintained in rule books and generally maintained separately from process documentation and software requirements. The following rule book types may be used to manage the quality and delivery of service to customers: • Governance • Service Delivery • Exception Processing • Compliance • Pricing
Touchpoints	A touchpoint represents one of the moments along the customer journey in which the customer interacts with the service provider or service, from first impressions to service retirement. Documenting touchpoints in the Service Architecture ensures the steps your customers take during the service lifecycle are carefully mapped and managed. Touchpoints are grouped into three different types: • Before Purchase (Brand and Marketing) • During Purchase (Service Acquisition) • After Purchase (Service Delivery)

Key Service Design Concept: Service Design Package

Another key concept in Service Design is ensuring we have the Service Design Package (SDP) completely documented. The Service Design Package follows a service through its lifecycle from initial proposal to retirement. It contains all the information required to manage an IT service. The SDP specifies the requirements from the viewpoint of the client (not IT) and defines how these are actually fulfilled from a technical and organizational point of view. Having the SDP documented in a central repository will...

- Improve the quality of services
- Improve decision-making
- Make implementation of new or changed services easier
- Improve alignment of services to the business
- Make service performance more effective
- Improve IT governance
- Make ITSM more effective
- Reduce Total Cost of Ownership (TCO)

Once a SDP is completed, it is passed from Service Design to the Service Transition phase to provide all information required to develop the service solution, including a preliminary (i.e., intended) time-schedule for the Service Transition phase. Service Transition, Operation, and Continuous Improvement provide input to the requirements in the SDP, ensuring services get better as time goes on.

According to ITIL, a Service Design Package (SDP) should consist of the following contents:

Requirements

This section includes the agreed and documented business requirements, such as the problem statement, vision, and business objectives. The requirements also include service contacts, such as the business stakeholders and customer representatives. These are the high-level details with which the rest of the SDP must align; you want to make sure to deliver the right service to the right group of people.

Service Design

Service Design refers to the functional requirements describing the new or changed service and the Service Level Requirements (SLRs), including service and quality targets. Also, this section includes the operational management requirements for the new or changed service (e.g., supporting services and agreements, control, measuring and reporting). Lastly, this section should document the plan for the service's transition, implementation, and operation. The Service Design section is very heavy in detail. In addition to the requirements for all service components and infrastructure, don't forget supporting processes and procedures, as well as measurements, metrics, and reports.

Organizational Readiness Assessment

The information in this section will make up the plan to assess benefit, financial, technical, and resource aspects, and needs to include a description of the new skills, competencies, and capabilities required to transition and implement the new or changed service.

Service Lifecycle Plan

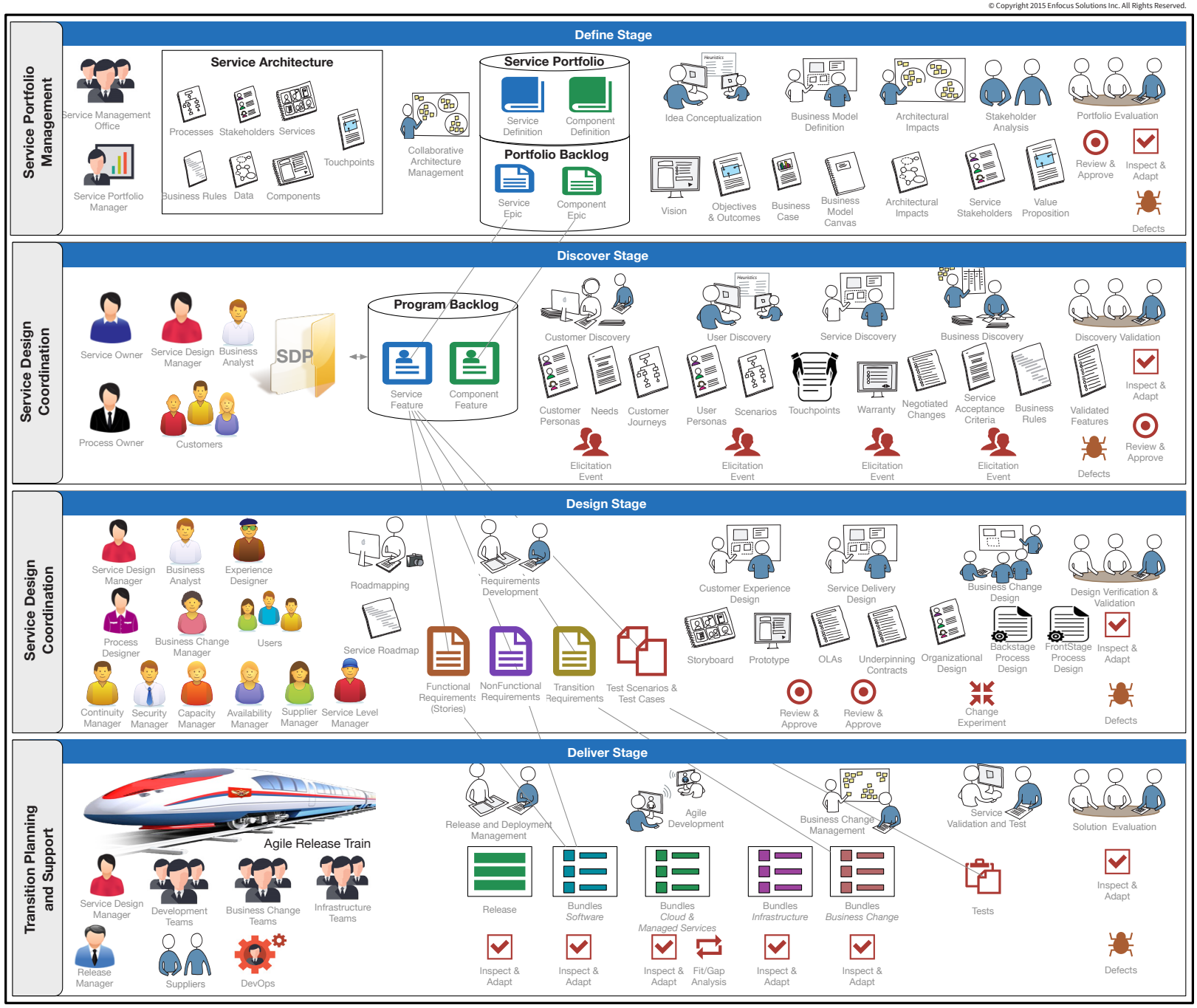
The last section includes the plan for each subsequent phase in the ITIL service lifecycle. This should include the required business processes, as well as a plan for communication and reporting. The Service Lifecycle Plan documents ways to manage related people, processes, and technology. It should also include timescales and quality targets for each phase. The Service Lifecycle Plan must include all details about service transition and operations focusing on the following:

- **Service Transition Plan**—document transition strategy, objectives, policies, risk assessment, and plans.
- **Service Operational Acceptance Plan**—document interface and dependency management, as well as planning, events, reports, and service issues regarding the new service and final service acceptance.
- **Service Acceptance Criteria**—document the acceptance criteria progression through each stage of the lifecycle.

Key Service Design Concept: Service Stakeholders

There are many stakeholders involved in the acquisition, delivery, and use of services. Gaining a good understanding of stakeholders and their needs and communicating what is expected of them is critical to providing quality service. The table below shows the various types of stakeholders for a service and how needs and expectations are documented for each stakeholder type.

SERVICE ELEMENTS	DEFINED BY	EXPECTATIONS
Customer	Value Proposition and SLA	• Customer Problem • Solution • Outcomes • Competitive Differentiation • Pricing Model
User	Task Analysis	• Performance Improvement • Removal of Constraints
Internal Supplier	OLA	• Services Provided • Value Contribution • Incident Management • Problem Management • Change Management • Service Exceptions • Performance Measurement
External Supplier	Underpinning Contract	• Services Provided • Value Contribution • Incident Management • Problem Management • Change Management • Service Exceptions • Performance Measurement
Service Delivery Team	Responsibilities	• Responsibilities • Value Contribution • Performance Measurement



The Four Stages of Service Design

Building a digital service is a complex task, with many risks. By breaking development into phases, we minimize risk, learning about what works and what doesn't, and iterating as you go.

As the service progresses through development, we'll find out more about users' needs, development requirements, and the conditions our service will be operating in. The level of detail, complexity, and risk will increase along the way. This approach allows the team who is creating and operating the service to start small, learn fast, and provide value to users as soon as possible. At Enfocus Solutions, we've broken the approach into four stages, which together make up the Agile Service Design Framework™. The four stages are described below and visualized in the diagram on page 10.

STAGE	DESCRIPTION
 **Define**	Before we start building a service, we need to create a picture of what the context for that service looks like. This involves identifying a market need and defining a solution concept to address that need. In this stage, financial viability is generally determined and documented in a business case; value is expressed in a value statement. These documents are presented to a service portfolio management committee who approves the expenditures for launching a new service or making enhancements to an existing service.
 **Discover**	The Discover Stage provides a more detailed understanding of user needs, what the existing service landscape looks like, and a sense of the features that the services may consist of by creating some initial prototypes and storyboards. The high-level business context will become clear, and we'll begin setting targets for your KPIs. We'll also get a better understanding of the legacy interfaces and infrastructure we must deal with, and what existing process are in place for replacing or decommissioning these. Discovery is about gaining an understanding of the problem or need. Many organizations create solutions before they understand the problem. Before creating the solution, we need to know what our market needs, wants, and is willing to pay for. Creating a solution before understanding the problem is not only a waste of time, but could cost our organization countless dollars and untold frustration when customers do not buy our service.

 **Design**	After needs are identified and understood, a solution is designed to meet those needs. This is generally done by defining Functional and Non-Functional Requirements to describe the utility and warranty for the service. The requirements are supplemented with storyboards, prototypes, and business change experiments to provide a good customer experience.
 **Deliver**	The Deliver Stage involves building the solution and validating that the solution provides the value specified in the value proposition and achieves the outcomes specified in the business case. In this stage, the technology components of services are designed using agile development teams. Also, solutions are tested using defined test scenarios and test cases, and validated against predefined conditions of satisfaction.

Each of the four stages of the Agile Service Design Framework™ has its own roles, activities, and artifacts.

1 Define Stage

Roles

Service Management Office

Similar to the Project Management Office, the SMO provides a center of excellence for delivering services. The SMO is a group or function within the organization that manages the definition and maintenance of standards for service management and related processes.

Service Portfolio Manager

According to ITIL, the Service Portfolio Manager decides on a strategy to serve customers in cooperation with the IT Steering Group, and develops the service provider's offerings and capabilities.

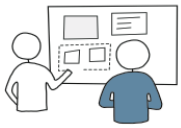
Activities

Collaborative Architecture Management

This activity involves the tasks to maintain the Service Architecture, explained on page 4.

Idea Conceptual-ization

Idea conceptualization is about identifying a market need and conceptualizing a solution to meet the need. Generally the need and solution are defined initially only as hypotheses and then expanded upon as various facts are discovered and assumptions are proven.



Business Model Definition

To deliver a new service, it is important to define the business model of how the service will be delivered. This is best done using a Business Model Canvas. The *Business Model Canvas* was originally developed by Alexander Osterwalder and is a strategic management and entrepreneurial tool that is extremely valuable in defining and designing services. Defining a business model for your service offerings requires requires defining the following:

Infrastructure—What resources, activities, and partners are needed to provide the service? This will be further elaborated with *architectural impacts*.

Service Offering—The services and products that the business offers, which meet the needs of its customers. According to Osterwalder, a company's *value proposition* is what distinguishes itself from its competitors. The business model will be further elaborated with...

- Service Definitions
- Value Propositions
- Touchpoints

Customers—To build an effective business model, a company must identify which customer segments it tries to serve. Various sets of customers can be segmented based on the different needs and attributes to ensure appropriate implementation of corporate strategy meets the characteristics of a select group of clients. To ensure the survival and success of any business, companies must identify the type of relationship they want to create with their customer segments.

A company can deliver its value proposition to its targeted customers through different channels. Effective channels will distribute a company's value proposition in ways that are fast, efficient, and cost effective. An organization can reach its clients either through its own channels (store front), partner channels (major distributors), or a combination of both. Definitions in the business model will be further elaborated with...

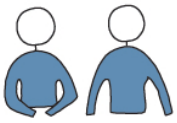
- Customer and user personas
- Use cases and scenarios
- Customer journeys and touchpoints

Finance Model—The business model must define the revenue streams, which are the way a company makes income from each customer segment. The cost structure defines fixed and variable costs for delivering the service.

Architectural Impacts

As epics are defined, it is important to understand the architectural impacts, gaps, and risks for developing a solution and transitioning from the current state to the future state. *Impacts* describe what needs to change. *Gaps* define the extent of the change due to moving from a current state to the future state. *Risks* describe what could go wrong when making the change. Understanding the impacts, gaps, and risks is critical for successfully deploying new software to the enterprise. Impacts, gaps, and risks are defined for the following areas:

- People
- Process
- Technology
- Data
- Rules



Stakeholder Analysis

Stakeholder analysis involves identifying who the customer is, who the users are, and who will be involved in delivering the service. Tools such as personas may be used to get an understanding of different types of stakeholders.

Portfolio Evaluation

Agile development generally works best for designing, building, and delivering services. The Scaled Agile Framework® (SAFe) breaks down development efforts into epics, features, and stories. Epics are managed at the portfolio level; features are managed at the program level; and stories are managed at the team level.

Similar to SAFe, the Agile Service Design Framework™ uses epics to initiate major strategic service initiatives such as chartering a new service, making major enhancements to an existing service, or retiring a service. The epics contain a lightweight business case, the service concept, the value proposition, and architectural impacts, and are presented to the portfolio management committee for Review and Approval.

Artifacts

Service Definition

A service definition includes a description of what the service does, who the customers of the service are, and the value delivered to the customer. A service definition consists of the following parts:

- **Service Concept**—The service concept shows how the organization serves its customers. From a clear and uniform sales process, we describe which services are offered, via which channels, and with which service level. Customer needs, brand values, and effectiveness are the relevant topics when defining the service concept.
- **Service Components**—The service definition defines what components are used to deliver the service.
- **Supporting Services**—Services that are not directly used by the business, but are required to provide other services. Examples include: hosting, directory services, naming services, etc. Service levels for supporting services are defined in an Operating Level Agreement (OLA).
- **Sourcing Model**—Describes the resources that are used to provide the service. Some services may be outsourced, some may be multi-sourced, and others may be provided totally using in-house resources.
- **Service Level Agreement**—An agreement between the service provider and the customer that describes the scope of the service, quality, and responsibilities for delivering the service. Generally, SLAs are defined for availability, security, capacity, and continuity.

Component Definition

Similarly to services, compontents must be designed and managed.

Components are defined in SO/IEC 20000-1:2011 as a "single unit of a service that when combined with other units will deliver a complete service." Examples include hardware, software, tools, applications, documentation, and support services.

ITI defines a *component* as a general term that is used to mean one part of something more complex. For example, a computer system may be a component of an IT service, and an application may be a component of a release unit.

ITIL uses a similar term known as a service asset. *Service assets* consist of resources and capabilities that a service provider uses and contribute to the delivery of the service. *Resources* could include infrastructure, applications, and data. *Capabilities* include people, organization, management, and their knowledge and wisdom. In essence, every single aspect of a service is considered a service asset.

According to ITIL, assets can be one of the following types:

- Management
- Organization
- Process
- Knowledge
- People
- Information
- Applications
- Infrastructure
- Financial Capital

When delivering a service, there are many ways to define components. A good component definition includes:

- **Capabilities**—The combination of processes, practices, tools, skills, and knowledge to deliver a specific outcome.
- **Applications and SaaS**—Software applications, either purchased, cloud-based, or customer written to deliver the service.
- **Infrastructure Technology and Services**—Services, storage, network, personal computing, and mobile devices needed to deliver the service.

Service Epic

A service epic documents the work of major changes to services, including chartering new services, making enhancements to services, and retiring services. Service epics are enterprise initiatives that are substantial enough in scope to warrant analysis and an understanding of potential return on investment (ROI) before implementation. Epics require a lightweight business case and impact analysis.

Component Epic

A component epic is similar to a service epic but is used to add, modify, or retire components. Purchased applications and managed services are good examples of epics. For example purchasing a new CRM system would be considered a component epic. Similar to a service epic, component epics can have features, associated requirements, needs, etc.

Vision

A vision is a description of the desired state or ultimate condition that should exist after the project has been completed. It is typically expressed in a vision statement, which is a clear and brief summary of what the project team members and their stakeholders expect to achieve. A good vision will describe the desired state and will also reference stakeholder interests, as well as provide relevant background.

Objectives & Outcomes

Business objectives are high-level business requirements that provide a clear understanding of the goals that the organization seeks to achieve. An objective can relate to a change desired by the organization or to a condition that the organization wants to maintain. Identifying business objectives contributes to the task of controlling scope creep and is the first step toward achieving a successful return on investment (ROI). Objectives and outcomes are defined for service and component epics to ensure that epics align with corporate strategy.

Business Case

A business case is used to document the business benefit and provide justification for an epic, and is used to support the related funding request for the epic. A business case compares the estimated resources, schedule, and costs of a project with the benefits that it is expected to provide. The business case must show that benefits outweigh costs. In many instances, it includes a financial analysis that calculates the project's ROI or the Net Present Value (NPV).

Business Model Canvas

The business model can be defined using nine building blocks:

1. **Customer Segments**—customers and others for which you are creating value
2. **Value Propositions**—how value is created for each customer segment
3. **Channels**—which touchpoints are needed to deliver value to a customer segment
4. **Customer Relationships**—definition(s) of the type of relationship(s) you are establishing with customers
5. **Revenue Streams**—how your business model generates revenue
6. **Key Resources**—the infrastructure required to generate value for the customer
7. **Key Activities**—the business processes used to deliver the service
8. **Key Partners**—suppliers and partners needed to leverage the business model
9. **Cost Structure**—a documented understanding of the cost structure (i.e., costs incurred) of the infrastructure required to deliver value

It is not enough to just enumerate the key building blocks; it is best to map them out using the Business Model Canvas, which helps define, plan, and create new business models.

Architectural Impacts

Whether a service or component epic, it will affect certain artifacts in the Service Architecture (refer to page 4). It's important to document these architectural impacts and manage related change throughout all stages of Service Design. If we can understand the impact of our service changes, we have a better chance of managing the change and gaining user adoption.

Service Stakeholders

Without the input of stakeholders, it's difficult to determine what the solution should help users achieve. The following types of stakeholders need to be identified and their roles and responsibilities understood:

- Customers
- Users
- Internal Service Suppliers
- External Service Suppliers
- Service Delivery Team
- Executives and Sr. Management
- Process Owners

Value Proposition

A *value proposition* provides two things: a promise of value to be delivered and acknowledged by the service provider, and a belief on the part of the customer that value will be delivered and experienced. Customer value propositions are important to ensure the service delivery, sales, and marketing teams can stay focused and aligned on what is important to the customer. Creating a value proposition is a part of developing a business strategy. As Kaplan and Norton say, "Strategy is based on a differentiated customer value proposition. Satisfying customers is the source of sustainable value creation."

Developing a value proposition is based on a review and analysis of the benefits, costs, and value that an organization can deliver to its customers, prospective customers, and other constituent groups within and outside the organization. It is also a positioning of value, where Value = Benefits – Cost.

2 Discover Stage

Roles

Service Owner

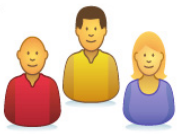
The Service Owner is the individual who owns a particular service, managing it through its complete lifecycle. The Service Owner ensures the service is managed with a business focus, and is responsible for managing continuous improvement.

Service Design Manager

According to ITIL, The Service Design Manager is responsible for producing quality, secure, and resilient designs for new or improved services. This individual is also responsible for producing and maintaining all design documentation.

Process Manager

The Process Manager is the individual who owns a particular process. According to ITIL, the Process Owner's responsibilities include sponsorship, design, and continual improvement of the process.



Customers

Customers are the people who buy IT services. The customer usually defines and agrees to the service level targets.

Business Analyst

The business analyst (BA) assists in elicitation of information from customers and users, facilitating customer, user, and service discovery activities. According to the IIBA, the BA acts as a liaison among stakeholders in order to understand the structure, policies, and operations of an organization, and to recommend solutions that enable the organization to achieve its goals.

Activities

Customer Discovery

Customer Discovery is a learning activity to determine who the customers are and what their needs are. This may be done through a wide variety of elicitation methods such as:

- Interviews
- Surveys
- Market Research
- Minimum Viable Products (MVPs)

The end goal is to answer the following questions:

- Who are the customers for the service?
- Why do they need the service and how will it deliver value to them?
- What are their service level requirements?
- What are the key touchpoints before purchase, during purchase, and after purchase?

User Discovery

User Discovery is a learning activity to identify the types of users of a service and how the service will support their work. The needs of the user are often different than that of the customer. A *persona* is prepared for the user types that are identified. *Use cases* or *scenarios* are prepared to document work or tasks that the user performs. This may be accomplished using a variety of elicitation methods such as:

- User Interviews
- Observation
- Day in the life of a user

The goal of User Discovery is to answer the following questions:

- What are the types of users that will use the service?
- What tasks do they perform and how will the service support those tasks?
- What are the challenges in getting users to adopt the service?
- What is needed to provide an excellent user experience?

Service Discovery

Service Discovery is an activity to discover what is required to deliver the service and provide an excellent customer experience. This will help to determine the set of features in the epic. It is used to answer questions such as:

- Will the service be insourced or outsourced?
- What resources and capabilities are needed to provide the service?
- What supporting services are needed to deliver the service?
- What are the KPIs or performance measures to evaluate service quality?
- What existing service assets can be used and what additions or changes are needed?
- What are the key touchpoints for the service that will deliver an outstanding customer experience?

Business Discovery

Business Discovery is a learning activity to identify what is needed from a business perspective to sell and deliver the service. It is important to examine customer touchpoints and how they relate to business processes and the skills and competencies of staff to provide an excellent customer experience. Business Discovery will answer questions, such as:

- What capability gaps exist in being able to deliver the service?
- Which business processes are needed to deliver the service?
- What business rules are required to govern the service to provide a consistent customer experience?
- What changes need to be made to frontstage and backstage processes to provide an excellent customer experience?
- What is the cost and pricing model for the service?
- Does the organization have the skills to provide the service?
- Given the service touchpoints, are frontline staff capable of delivering an outstanding customer experience?

Discovery Validation

The purpose of Discovery Validation is to validate that features are needed and will deliver value to the customer. According to Standish Group Research, 64% of functionality developed is rarely or never used. This is the result of two factors: 1) producing software without validating that there is a real need, and 2) lack of user adoption due to insufficient attention being placed on the user experience.

Agile is fantastic for developing quality software. It can significantly improve time to market, quality, and customer satisfaction. However, it is not so good for discovering the right product to build, and discovering what the customer really needs or validating the right market. Nor is agile a very good solution for IT departments that need to understand what the real business problem is and to develop solutions to improve business outcomes. These activities are discovery activities and require a different type of skill and a different method. Separating Discovery from Delivery yields big benefits for most organizations. *Discovery* is about quickly generating validated product backlog items, and the *Delivery* will be focused on generating releasable software. An effective Discovery Validation process will result in…

- Elimination of features that provide little or no value
- Better user experience
- Better business outcomes
- Less rework
- Effective business change
- More cost effective validation by avoiding writing code to validate need

Artifacts

Service Feature

A feature is a common term that service managers, marketing, and sales personnel typically use to describe what the service does and the benefits it provides. Features are central to Service Design. They bridge the gap for Discovery (i.e., understanding the needs of the users and stakeholders) to Delivery (i.e., software and processes designed to meet those needs). When defining features, it is also important to define the benefits they will provide to the customer. Discovery activities are managed with features using a Feature Roadmap. Features are also used to manage Delivery activities as they are generally tied to a specific release.

Component Feature

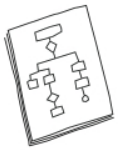
Component Features are very similar to Service Features, except they represent the building blocks of components.

Needs

A Stakeholder Need Pattern is a guide to writing a particular type of need. Patterns are created to explain how to approach each type of need, what to enter into each field, and what to worry about. Need Patterns help users write stakeholder needs to be used as a basis for solution requirements by pointing out the details that must be focused on. This type of artifact ensures we understand the point of view of the customer/user because we have their needs written in their own words.

Customer Personas

Personas document key assumptions and information about important individuals, such as stakeholders, customers, users, etc. Customer Personas specifically document the needs, desires, and profile of the service provider's customers in the form of a narrative. Customer Personas are useful in understanding the best ways to present services to our customers.

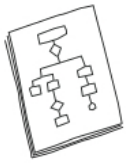


Customer Journey

A customer journey includes the full customer experience, including all interactions between the customer and the service provider. It includes how the customer becomes aware of our brand, their first interactions, and all subsequent communications.

User Personas

Personas document key assumptions and information about important individuals, such as stakeholders, customers, users, etc. User Personas specifically document the goals and behaviors of a service's users in the form of a narrative. User Personas are useful in discovering the features and functions that will not only satisfy but delight your users.



Scenarios

Written from the point of view of a user, scenarios are descriptions of the specific actions the user performs to accomplish a goal. There are three types of scenarios:

- **Activity**—a description of an activity that is or will be commonly performed.
- **Interaction**—a description of an interaction between two entities, either individuals or software interfaces.
- **Problem**—a description of a problem that has occurred.

A major benefit of a scenario is its ability to be used and grasped by people without a specialized background. Therefore, we are able to easily use scenarios in design activities that involve user participation. As such, these scenarios are beneficial when used to convey an interaction from the perspective of an end user.

Touchpoints

A touchpoint is any instance along the customer journey in which a customer interacts with the service or service provider.

Warranty

According to ITIL, half of the value of a service is in its warranty, also known as fit for use, meaning the service meets certain non-functional requirements, like that it is available at the right time. The four major areas of warranty are availability, continuity, security, and capacity.

Negotiated Changes

After a service or component definition has been approved, any necessary changes must be documented and negotiated with the appropriate stakeholders.

Service Acceptance Criteria

According to ITIL, Service Acceptance Criteria is a set of criteria used to ensure that an IT service meets its functionality and quality requirements, and that the IT service provider is ready to operate the new IT service whenever it gets deployed. Developing the Service Acceptance Criteria provides us with a list of exactly what needs to be tested before deployment.

Business Rules

Business rules describe in plain language policies for making decisions, formulas for calculations, definitions used in the business, and key facts and assumptions of how the business operates. Business rules are generally owned by the business and not IT. There are five types of business rules: Terms, Facts, Constraints, Action Enablers, and Calculations. During the Discover Stage, we'll review the business rules in the Service Architecture to determine whether any need to be changed, retired, or created.

Validated Features

During the Discover Stage, the feature backlog is validated to ensure the solution will address customer needs in an achievable timeframe. To do this, business owners will complete the Review and Approval process.

3 Design Stage

Roles

Service Design Manager

According to ITIL, the Service Design Manager is responsible for producing quality, secure, and resilient designs for new or improved services. This individual is also responsible for producing and maintaining all design documentation.

Business Analyst

The IIBA defines the business analys (BA) as an agent of change and a liaison among stakeholders who fully understands the structure, policies, and operations of an organization, and is able to recommend solutions that enable the organization to achieve its goals.

Experience Designer

The Experience Designer is the individual or team responsible for ensuring the service's user experience conforms to user expectations. This role is involved in the design and continuous improvement of the service and its processes.

Process Designer

The Process Designer is responsible for designing the frontstage and backstage processes needed to deliver the service. Process designers often build process models using Business Process Modeling Notation (BPMN). Also, simulation tools are often used to ensure process designs are efficient and deliver expected results within expected cycle times.

Business Change Manager

The Business Change Manager ensures benefits are realized by managing changes to services.

Users

Users are people who use the services delivered by the organization. Users are different than customers, as customers do not always use the service.

Continuity Manager

According to ITIL, the Continuity Manager is responsible for managing risks that could seriously impact IT services. He ensures that the IT service provider can provide minimum agreed-to service levels in cases of disaster, by reducing the risk to an acceptable level and planning for the recovery of IT services.

Security Manager

The Security Manager is the individual responsible for ensuring the confidentiality, integrity, and availability of an organization's assets, information, data and IT services. According to ITIL, the Security Manager also establishes an enterprise security stance through policy, architecture, and training processes.

Capacity Manager

The Capacity Manager is the individual responsible for ensuring a service has the capacity to deliver agreed upon performance targets.

Availability Manager

The Availability Manager is the individual responsible for managing all aspects of the availability of a service. According to ITIL, the Availability Manager is responsible for ensuring that all IT infrastructure, processes, tools, roles, etc. are appropriate for the agreed service level targets for availability.

Supplier Manager

The Supplier Manager is the individual responsible for managing suppliers and ensuring value for money is obtained. The Supplier Manager is responsible for developing contracts with suppliers that support business needs and ensuring suppliers meet their contractual requirements.

Service Level Manager

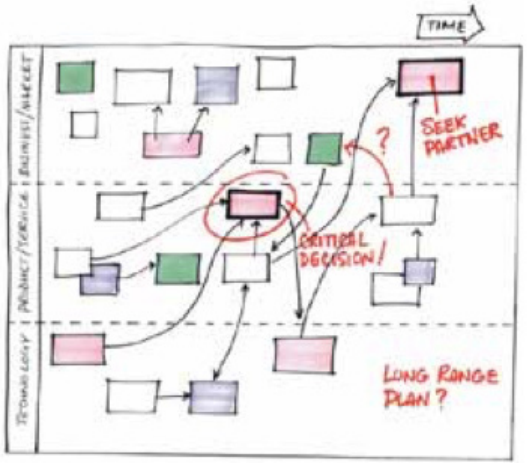
The Service Level Manager is the individual responsible for negotiating Service Level Agreements (SLAs) and ensuring services are designed in accordance with the agreed service level targets. The Service Level Manager also ensures the OLAs and underpinning contracts are appropriate in the context of the SLA.

Activities

Roadmapping

A *roadmap*, or a strategic plan with long- and short-term goals, can mean the difference between success and failure when planning, delivering, and marketing your service. They can be the most critical strategic planning tool used by service management. Roadmaps can win and keep customers, guide engineering, and set the vision and expectations for everyone.

Roadmapping is actually a simple process; it involves identifying what features will be delivered to the customer and specifying a timeframe for when they will be available. The end result is a Service Roadmap. It is a very flexible approach that can be customized to address the particular strategic needs and issues facing firms, government agencies, and other organizations. No two roadmapping exercises are the same, but workshops typically form a key part of the process. Workshops bring together a group of experts and, using a set of graphical templates, help them visualize and develop strategies.



Requirements Development

This activity is the development of a set of requirement specifications for each validated feature. Different types of requirements are needed depending on the situation, as described on page 29. Requirements should be developed using Requirement Patterns to ensure all necessary information is included in the requirement.

Customer Experience Design

Customer Experience Design is performed by first defining customer journeys and touchpoints. Each key touchpoint must be carefully designed to achieve an outstanding customer experience. Prototypes and storyboards are often prepared for key digital touchpoints. Business rule books are often prepared for non-digital touchpoints to ensure consistency of service and exception handling.

Service Delivery Design

Service Delivery Design involves planning of the capabilities, resources, and supporting services necessary to provide a seamless customer experience. This often involves drafting Operating Level Agreements (OLAs) or Underpinning Contracts when external suppliers are needed.

The goal of Service Level Management is to understand the requirements of the customer and organization, factor in the capabilities of the supplier(s), and then deliver quality services that meet those requirements and are subject to constant improvement. This entire process is orchestrated by Service Delivery Design.

Service Delivery Design also involves defining Service Level Requirements for the service, which are usually in the form of Non-Functional Requirements. Service Level Requirements are defined at a minimum for:

- Availability
- Capacity
- Security
- Continuity

Business Change Design

IT is increasingly changing the way customers receive value, directly through IT-enabled products and services, and indirectly through more efficient product/service development, production and delivery, and customer support. Delivering superior customer value is the true purpose of the enterprise and everyone in it. Yet in survey after survey, the majority of business people report that IT does not understand their business, understand customer needs, or deliver value proportional to the investment made in IT. Frequent complaints include that the IT organization is slow to respond, engages in projects that rarely finish successfully or on time, creates systems with excessive complexity that are difficult to use and maintain, and is unable to keep up with the rapid pace of business change. These surveys repeatedly show that executives are often "baffled, frustrated, and even angered by their IT organizations."

In response, many organizations have begun to adopt agile methods for software or product development. Agile methods have helped organizations deliver more rapidly, increase customer satisfaction, and improve quality. However agile development alone does not make the enterprise agile. An agile business must be able to rapidly design business changes that affect people, processes, data, technology, and rules to support threats and opportunities in the market.

Design Verification & Validation

Many companies experience the pain of expensive recalls, costly product rework, and unexpected delays in product or service releases. A major contributing factor is the lack of an effective process of design verification and validation. Ensuring your customers' needs are met as the design progresses is a critical part of developing a service, and can help you avoid these expenses and delays.

ISO 9001 is one of the standards within the range of ISO 9000 quality management standards. Probably the most misunderstood concept in ISO 9001 is the difference between *Design Verification* and *Design Validation*. These two steps are distinctly different, and important in a good design process. One step is used to make sure that the design has addressed every requirement, while the other is used to prove that the design can meet the requirements set out for it.

Each of these steps is important in the design process because they serve two distinct functions. Verification helps ensure that no requirements are missed in the design, whereas validation focuses on whether the designed solution will achieve the business objectives, deliver what the customer wants, and achieve the outcomes expected. Together, they ensure that the product designed will satisfy the customer needs, which is one of the key focuses for ISO 9001 and improving customer satisfaction.

Artifacts

Service Roadmap

A service roadmap is a plan of long and short term strategies and goals for the initial creation and continuous improvement of the service, which includes a description of which features will be delivered at what time.

Functional Requirements (Stories)

Functional Requirements contribute to the physical design of the system. According to *The BABOK Guide*, Functional Requirements describe the behavior and information that the solution will manage. This type of requirement specifies functionality that the developers must build into the system to enable users to accomplish desired tasks, thereby satisfying Business Requirements or Objectives.

Non-Functional Requirements

Non-Functional Requirements describe the operation or performance of the system, contributing to the system architecture. According to *The BABOK Guide*, Non-Functional Requirements capture conditions that do not directly relate to the behavior or functionality of the solution, but rather describe environmental conditions under which the solution must remain effective or qualities that the system must have. A Non-Functional Requirement states how the system must perform, but there is no way to physically build the requirement.

Transition Requirements

Transition Requirements do not address the solution, but rather the enterprise-wide transition to the solution. According to *The BABOK Guide*, Transition Requirements describe capabilities that the solution must have to facilitate a successful transition from the current state of the enterprise to a desired future state. However, these capabilities will not be needed once transition is complete. Contrast to Functional and Non-Functional Requirements, Transition Requirements are not commonly reused.

Test Scenarios & Test Cases

A *test case* focuses on a single task to be performed by the user. A *test scenario* consists of a group of related test cases. There should be at least one test case created per requirement. For maximum efficiency, test cases are created alongside requirements.

Storyboard

The storyboard is a tool derived from the cinematographic traditions; it is the representation of use cases through a series of drawings or pictures, put together in a narrative sequence. It is a visual representation in the form of illustrations or images displayed in sequence for the purpose of visualizing how a service works before building it.

The service storyboard shows the manifestation of every touchpoint and the relationships between the service and the user in the creation of the experience.

Prototype

The service prototype is a tool for testing the service by observing the interaction of the user with a prototype of the service put in the place, situation, and condition where the service will actually exist.

The aim is to verify what happens when some external factors interfere during the service delivery, factors that it's not possible to verify during the Discover stage, but have a great impact on the user perception and experience.

OLAs

An Operational-Level Agreement (OLA) defines the interdependent relationships among the internal support groups of an organization working to support a Service-Level Agreement (SLA). The agreement describes the responsibilities of each internal support group toward other support groups, including the process and timeframe for delivery of their services. The objective of the OLA is to present a clear, concise, and measurable description of the service provider's internal support relationships.

Underpinning Contracts

Underpinning Contracts are written contracts between the service provider and other external service providers. They are very similar to an OLA, but are also legal contracts.

Organizational Design

The Organizational Design defines the roles and responsibilities for delivering the service. It is often presented in the form of a RACI matrix.

Backstage Process Design

Backstage processes are business processes that are performed in the background and have no direct line of interaction with the customer. Examples of backstage processes may include accounting, training and other human resource management processes, and infrastructure management.

Service management and design has largely focused on the interactions between employees and customers. This perspective holds that the quality of the service experience is primarily determined during the final service encounter that takes place in the frontstage. This emphasis discounts the contribution of the activities in the backstage of the service value chain where materials or information needed by the frontstage are processed.

However, the vast increase in web-driven consumer self-service applications and other automated services requires new ways of thinking about service design and service quality. It is essential to consider the entire network of services that comprise the back and frontstages as complementary parts of a service system. We need new concepts and methods in service design that recognize how backstage information and processes can improve the front stage experience.

Frontstage Process Design

Frontstage processes are business processes in which there is direct interaction with the customer. For person-to-person services, a central idea is that the quality of the service is determined in the service encounter at the moment of truth when the service is delivered or "co-produced." Most consumers of person-to-person services expect some flexibility or customization because limited choices can give a service a transactional, or take it or leave it, character that customers perceive as a low quality experience. Therefore, an important concept in Service Design is to empower the service provider/employee to adapt the service or provide additional services to solve problems or handle unexpected events, or just so that the customer can have it his/her way. This view treats variability in service delivery as inevitable and perhaps even desirable; however, it makes Service Design difficult.

4 Deliver Stage

Roles

Agile Release Train

An Agile Release Train is a large group that consists of multiple agile development teams, which follow a common cadence but have their own dedicated resources.

Release Manager

The individual responsible for managing the development and delivery of a software release. According to ITIL, the Release Manager is responsible for planning and controlling the movement of releases to test and live environments.

Development Teams

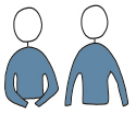
The development team is a cross-functional unit working together with the necessary abilities to build, test, and deploy.

Business Change Teams

Business change teams are responsible for managing business change. This involves developing new skills to provide a service, changing business processes to support the service, and designing business rules to support the service.

Infrastructure Teams

Infrastructure teams make necessary changes to the technology infrastructure to ensure stated service level requirements are met.



Suppliers

According to ITIL, suppliers are third-parties responsible for supplying goods or supporting services required to deliver one of our services.

DevOps

DevOps ensures communication, collaboration, and cooperation between the development teams and the business.

Service Design Manager

In the Deliver Stage, the Service Design Manager works closely with the Release Manager to ensure that the new or changed solution delivers value to the customer.

Activities

Release and Deployment Management

Prior to agile, releases were usually predefined by a set of requirements documents consisting of a Business Requirements Document (BRD), Systems Requirements Specifications, and a System Design Document, which collectively defined the functionality for the release. In most situations, there was a single "big bang" release that happened at the end of the project. The agile approach differs significantly, as the goal is to deliver a continuous stream of value released to the customer via many deployments while the software is constantly developed. In agile the release question becomes, "When do we have sufficient new functionality (the right batch size, for us and our customer) to warrant release?" This question is answered not once, but continuously. This creates the continuous flow of value.

Many organizations have adopted Release and Deployment processes based on ITIL®. These processes often need to be redesigned to support agile development and release management.

Agile Development

Agile development provides opportunities to assess the progress throughout the development lifecycle. This is achieved through regular cadences of work, known as Sprints or iterations. At the end of each iteration, teams present a potentially shippable product increment. By focusing on the repetition of abbreviated work cycles as well as the functional product they yield, agile methodology is described as "iterative" and "incremental." In waterfall, development teams only have one chance to get each aspect of a project right. In an agile paradigm, every aspect of development — requirements, design, etc. — is continually revisited. When a team stops and re-evaluates the direction of a project every two weeks, there's time to steer it in another direction.

This "inspect-and-adapt" approach to development improves quality and reduces time to market. Because teams can develop software at the same time they're gathering requirements, "analysis paralysis" is less likely to impede a team from making progress. And because a team's work cycle is limited to two weeks, stakeholders have recurring opportunities to calibrate releases for success in the real world. Agile development helps companies build the right product or solution.

Scrum is the most popular agile method due to its simplicity and flexibility. However, many organizations that claim to be doing Scrum, aren't doing anything close to Scrum's actual definition.

Scrum emphasizes empirical feedback, team self management, and striving to build properly tested product increments within short iterations. Doing Scrum as it's actually defined often conflicts with the existing culture of non-agile organizations.

Business Change Management

When running a change management program, we try to help the organization get to improved business outcomes. We do this by defining a target state and planning a set of change actions. Many of the upfront choices we make concerning our change are really just assumptions. As we execute our change plan, we continually uncover new information about business value, existing capabilities, current culture, workload, and a variety of other facts. This new information requires us to constantly rethink the validity of our assumptions.

Existing change management methods make it really hard to ensure that our change plan keeps up with our continued learning. In most cases, major failure has to occur before a change in direction is considered. As a result, organizations end up with a change that does not provide the intended value. In the Lean Change Method, changes are negotiated with the stakeholders that will be impacted by the change and assumptions are tested by defining Minimal Viable Changes (MVCs) and tested through a series of experiments.

In the Lean Change Method, MVCs are introduced to the organization using a validated change lifecycle. The Validated Change Lifecycle integrates Kotter's Eight Steps with the Meta-Iteration Lifecycle Pattern from the book, *Running Lean*. Using this lifecycle, MCVs are both defined and validated according to a specific sequence as shown in the diagram below.

Even though the Lean Change Method was primarily designed for organizational change, the same concepts can also be applied to other areas of change such as:

- Data
- Business Processes
- Technology
- Business Rules

For more on change management and Service Design, refer to page 41.

Service Validation and Testing

Service Validation and Testing was introduced as a new process in ITIL V3. The objective of Service Validation and Testing is to ensure that deployed releases and the resulting services meet customer expectations, and to verify that IT operations is able to support the new service.

Solution Evaluation

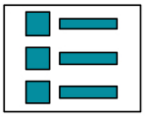
Solution Evaluation is a knowledge area in the *Business Analysis Body of Knowledge* (*BABOK Guide*) V3. The Solution Evaluation Knowledge Area is comprised of the following tasks:

- **Measure Solution Performance**—determine the most appropriate way to assess the performance of a solution, including how it aligns with enterprise goals and objectives, and collects information on solution behavior.
- **Analyze Performance Measures**—examine information regarding the performance of a solution in order to understand the value it delivers to the enterprise and to stakeholders, and determines whether it is meeting current business needs.
- **Assess Solution Limitations**—investigate issues within the scope of a solution that may prevent it from meeting current business needs.
- **Assess Enterprise Limitations**—investigate issues outside the scope of a solution that may be preventing the enterprise from realizing the full value that a solution is capable of providing.
- **Recommend Actions to Increase Solution Value**—identify and define actions that the enterprise can take to increase the value that can be delivered by a solution.

Artifacts

Release

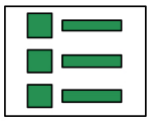
According to SAFe, a release is an event in which the user gets the full benefits of the developed solution. The solution has been designed, developed and tested, and now there is a potentially shippable product increment ready.



Bundles

(Software)

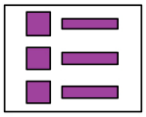
Traditionally, all software requirements are placed in a single requirements document. This archaic process does not support agile development, nor is it efficient for managing and coordinating the work of various teams involved in building the service. Bundles are groupings of requirements used in place of a requirements document that are assigned to teams to be completed within a given timeframe. In agile, the timeframe would usually be called Sprints.



Bundles

Cloud & Managed Services

For many services, external service providers or cloud-based services are used to provide the service. Requirements for these items are placed in a separate bundle. Often a fit/gap analysis is performed to compare one supplier's offerings to another to select the best alternative for the service.



Bundles

Infrastructure

Often providing a new service requires new or additional hardware. Infrastructure requirements are placed in a separate bundle to more effectively manage changes to infrastructure.

Recurring Events in Service Design

Service Design activities and processes should be continuously optimized through a Continuous Service Improvement Process. At Enfocus Solutions, we document continuous service improvement through the definition and management of Lifecycle Events, or events that occur throughout the lifecycle of an epic, which are explained below.

LIFECYCLE EVENTS	OCCURS DURING	DESCRIPTION
 **Elicitation Event**	Discover Stage	It is essential to understand the customer and other stakeholder needs when developing a new service or making changes to an existing service. The process of obtaining this knowledge is referred to as elicitation. A major goal of *elicitation* is to avoid the confusions between stakeholders and the Service Design team. There are a wide variety of elicitation methods; examples include: • Interviews • Observation • Documentation Review • Voice of the Customer Surveys • Focus Group • Requirements Workshop • Conversations • Surveys • Questionnaires • Brainstorming Sessions • Interface Analysis • Reverse Engineering • Contextual Inquiries

 **Inspect & Adapt**	Define Stage Discover Stage Design Stage Deliver Stage	At key intervals, it is important to reflect on how well things are going and make necessary changes when problems arise. Teams demonstrate the current state of the solution and obtain quick feedback from stakeholders to make needed course corrections. Examples of Inspect and Adapt activities include: • Sprint Retrospective • Solution Demo • Design Review • Strategy Review • Performance Review • Process Review • Inspect and Adapt Workshop
 **Review & Approve**	Define Stage Discover Stage Design Stage	Many items require formal customer or management Review and Approval. This process is done by reviewing artifacts such as: • Roadmap • Feature Definition • Service Definition • Service Level Agreements • Operating Level Agreements
 **Tests**	Deliver Stage	Thorough testing must be performed before delivery is completed. The three types of tests that must be performed include: • System Test • Acceptance Test • Performance Test
 **Change Experiment**	Design Stage	In today's fast pace environment, it is critical that organizations develop a competency in business change management. Traditional change methods simply do not work; many recent studies show that over 70% of change efforts fail. New methods such as the Lean Change Method are beginning to emerge where changes are negotiated and tested using change experiments before changes are rolled out to the organization. It is important in a service environment to code services, negotiate changes, and validate potential changes before implementation. Read more about change management on the next page.

Change Management and Service Design

To maximize the value delivered by services, it is essential not only to manage the services themselves, but also to manage how the business or customer utilizes the services. For services offered to external customers, it is vital to align and integrate business processes with IT services to provide a seamless flow of value to the customer. The goal of *business change management* is to manage how fast the business can assimilate necessary change to realize benefits from new features offered in the service. Below is a list of problems that many organizations face, which prevent the organization from achieving maximum value from IT service offerings:

- Business changes are not managed and simply left up to the business units.
- Organizational change methods are old and based on waterfall practices.
- Real users are not involved in defining requirements and testing.
- Software changes are made without considering the business process.
- Cultural issues.
- Transition requirements are not defined.
- Business changes are not adequately considered.
- Business change management is not aligned with software development.

It is important to adopt new change management methods when agile development is used to provide a continuous set of features to customers. Traditional organizational change methods simply do not work in an agile environment where services are continuously improved and a constant stream of service improvements are being delivered to the customer. One of the best methods for implementing organizational change in the age of the customer is the Lean Change Method. The Lean Change Method involves:

- Building a change canvas with the change recipients
- Negotiating the change strategy documenting risks, assumptions, and the hypothesis
- Planning change experiments using MVCs to test the hypothesis and assumptions
- Pivoting when hypotheses and assumptions do not check out
- Designing roll-out and accelerate implementation when change experiments are proved
- Measuring performance and adapt accordingly

Service Design Tools

There are many design methods that can be used to visualize and communicate design requirements for services. One of the best sources for information on Service Design tools is ServiceDesignTools.org. The website is the result of the research activity done by Roberta Tassi during her graduation thesis. Service Design Tools was conceived as an open platform of knowledge that was intended to be shared with the design research community. The diagram below, taken from the website, shows various tools that are described on the site.

Additional Information: Business Analysis Core Concept Model™

The Business Analysis Core Concept Model™ was introduced in Version 3 of the *Business Analysis Body of Knowledge* (*The BABOK Guide)*. The Business Analysis Core Concept Model (BACCM) is a conceptual framework for business analysis. BACCM is based on six core concepts: Change, Solution, Context, Value, Stakeholder, and Need. Each core concept is an idea fundamental to the practice of business analysis. All the concepts are equal and necessary; there is no prime concept; they are all defined by the other core concepts. The BACCM is fundamental to business analysis and can be applied to Service Design. A diagram of the BACCM is shown below.

The *BABOK Guide* defines each of the terms as shown in the table below.

CONCEPT	DEFINITION
Change	A controlled transformation of the enterprise.
Need	A problem, opportunity, or constraint with potential value to a stakeholder.
Solution	A specific way of satisfying one or more needs in a context.
Value	The worth, importance, or usefulness of something to a stakeholder within a context.
Stakeholder	A group or individual with a relationship to the change or the solution.
Context	The circumstances that form the setting for a change and allows for further understanding and assessment of the change.

By answering the following questions, you can determine if you have completed the stage activities and artifacts with the core concepts in mind, leading to higher quality analysis and maximizing the value to your stakeholders.

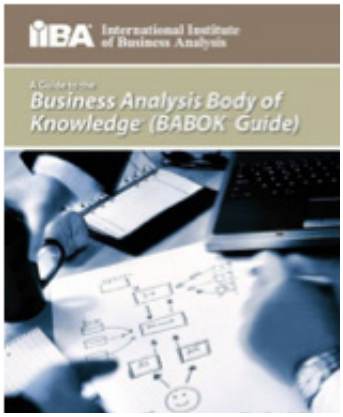
- What is the kind of change we're doing?
- What are the needs we're trying to satisfy?
- What are the solutions we're creating or changing?
- Who are the stakeholders involved?
- What do stakeholders consider to be of value?
- What is the context that we and the solution are in?

If any of the core concepts experience a change, it should cause us to reevaluate these core concepts and their relationships to value delivery. Replacing "Solution" with "Service" and "Context" with "Market" makes an excellent model for Service Design. For more information on the BACCM, please refer to the *BABOK Guide V3*.

Additional Information: References and Sources that Inspired the Agile Service Design Framework™

ITIL® is the most widely accepted best practice for IT Service Management. ITIL® was originally developed by the UK Government but is now managed and supported by Axelos.

The *Service Strategy* manual provides instructions on how to position and define services as strategic assets. The principles described in *Service Strategy* underpin the service lifecycle with helpful policies and guidelines as well as processes in the context of the lifecycle phases of service design, service transition, service operation und continual service improvement.



The *Business Analysis Body of Knowledge* (*BABOK Guide*) was developed by the International Institute of Business Analysis. It consists of six knowledge areas:

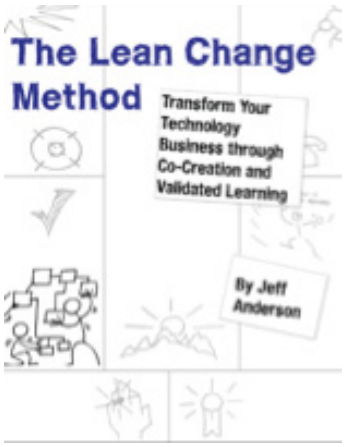
- Business Analysis Planning and Monitoring
- Elicitation and Collaboration
- Requirements Lifecycle Management
- Strategy Analysis
- Requirements Analysis and Design Definition
- Solution Evaluation

ITIL® is the most widely accepted best practice for IT Service Management. ITIL® was originally developed by the UK Government but is now managed and supported by Axelos.

The *Service Design* manual provides guidelines for the design and the development of services and service management processes. The volume includes design principles and methods for the realization of strategic goals in service portfolios and service assets. However, the scope of *Service Design* is not limited to new services. It also contains advice on necessary changes and improvements to enhance or maintain the added value of the services across the individual lifecycles, ensure their continuity, achieve the service levels and fulfill the compliance requirements. Service Design provides the organization with valuable advice on the question of how the design capabilities for service management can be developed and acquired.

Scaled Agile® Framework

The Scaled Agile Framework® was developed by Dean Leffingwell. This model of agile adoption has been elaborated primarily in his books *Agile Software Requirements: Lean Requirements for Teams Programs and the Enterprise* (2011) and *Scaling Software Agility: Best Practices for Large Enterprises* (2007). The framework has been successfully applied in programs of only 50-100 people, and in enterprises employing thousands of software developers.



The Lean Change Method itself is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License to Jeff Anderson and Alexis Hui, based on the work that can be found at http://leanchangemethod.com.

The method is based on concepts from the Lean Startup method and Kotter's eight step lifecycle. The work provides an excellent method for effecting organizational change in a lean-agile environment.

The Lean Startup by Eric Ries provides a scientific approach to creating and managing startups and get a desired product to customers' hands faster. *The Lean Startup* method teaches you how to drive a startup—how to steer, when to turn, and when to persevere—and grow a business with maximum acceleration. It is a principled approach to new product development. It is based on the following principles:

- Entrepreneurs are everywhere
- Entrepreneurship is management
- Validated Learning
- Innovation Accounting
- Build-Measure Learn

Value Proposition Design was written by Alex Osterwalder, also the creator of the Business Model Canvas. Value Proposition Design helps organizations design and create products and services that customers want.

About Enfocus Solutions Inc.

Enfocus Solutions is a provider of software and services to help organizations achieve business agility and deliver more value to customers with IT services and products. Enfocus Solutions' software and services enable organizations to implement agile best practices at all levels in the organization, ensuring customer needs are met. Enfocus Solutions fully supports the agile definition and maintenance of end-to-end IT and shared services, providing the capabilities to successfully enable change in the organization and achieve higher ROI.

Contact Information

Email: info@EnfocusSolutions.com

Phone: (210) 399-4240

Toll-free: (877) 253-0275