High Velocity ITSM

AGILE IT SERVICE MANAGEMENT FOR RAPID CHANGE IN A WORLD OF DEVOPS, LEAN IT AND CLOUD COMPUTING

Type of Waste	Description
Inventory	Excess products and materials that are not being used
Talent	Improper or inefficient use of people skills and knowledge
Waiting	Wasted time waiting for the next step in a process
Motion	Unnecessary movement of people
Defects	Efforts to fix data errors, program bugs or other types of failures
Transportation	Unnecessary movement of data or outputs
Overprocessing	Producing at levels of quality more than required by the customer
Overproduction	Creating more output than what is needed or before it is needed

Randy A. Steinberg

Table of Contents

Other books by Randy A. Steinberg:

Implementing ITSM
Adapting Your Organization to the Coming Revolution in IT Service Management
Trafford Press ISBN: 978-1-4907-1958-0

Measuring ITSM
Measuring, Reporting and Modeling the IT Service Management Metrics
That Matter Most To IT Senior Executives
Trafford Press ISBN: 978-1-4907-1945-0

Servicing ITSM
A Handbook of IT Services for Service Managers and IT Support Practitioners
Trafford Press ISBN: 978-1-4907-1956-6

Architecting ITSM
A Reference for Architecting and Building the Entire IT Service Management Infrastructure End To End
Trafford Press ISBN: 978-1-4907-1957-3

Organizing ITSM
Transitioning the IT Organization from Silos to Services With Practical Organizational Change
Trafford Press ISBN: 978-1-4907-6270-8

The average change request takes 9-22 business days to get an approval
35% of IT organizations report 1-3 major outages per week
70% of IT budgets are stuck in non-discretionary spend just to keep the wheels running
85% of IT organizations get blind-sided when new services go live
60% of IT organizations experience unplanned labor rates of 35% or higher
80% of IT organizations cannot clearly articulate their services to the business
6 out of 10 applications fail because they cannot be operated at acceptable cost and risk

Let’s turn IT into a LEAN and EFFICIENT Service Organization!

- The Author

Dedication

This book is dedicated to those very hard working IT professionals, managers and executives who deserve to see their IT solutions deploy and operate day-to-day within acceptable levels of costs and risks to their company.

Chapter 1

An ITSM High Velocity Overview

The Landscape Is Changing

IT is undergoing a fundamental change in how it manages and operates itself.

Without our realizing it, the primary role of IT as an engineering organization has been shifting from a primary focus on building solutions to one of integrating solutions from internal and external providers. From engineering to service integration. The practice of understanding business requirements, designing building and deploying IT solutions is giving way towards integrating pieces, parts and services together to meet those same business requirements. IT executives that don’t understand this shift will face severe challenges in meeting business needs at cost and availability levels acceptable by the business.

The traditional IT operating model of delivering IT to the business in the form of bundled capabilities and assets is now wearing thin in an age of cloud computing, on-demand services, virtualization, outsourcing and rapidly changing business delivery strategies. The role of IT is rapidly changing from technology tinkerer to services thinker. Traditional alignment approaches are no longer enough. IT must blend with the business.

Many IT organizations have yet to recognize that the traditional IT operating model of delivering IT to the business in the form of bundled capabilities and assets is now wearing thin in an age of cloud computing, on-demand services, virtualization, outsourcing and rapidly changing business delivery strategies. The role of IT has now changed from engineer-to-order to service integrator. What IT traditionally engineered, built, owned and operated can now be bought from many sources more easily without inheriting the specific risks of ownership, support, building and managing an operating infrastructure.

The IT executive that cannot clearly articulate the services they deliver, the IT spend for those services, and how those services are consumed by the business is essentially running a commodity operation that is whip-sawed by business demand and will never been seen as nothing more than an overhead cost to the business.

At the time this book is being written, many IT organizations are still organized and operated as siloed engineering delivery and capability organizations. If the organization chart looks like the schema for a configuration database (e.g. application unit, network unit, server unit, etc. all reporting to an executive layer) then that organization may be in trouble. The forces of outside suppliers ready to provide hardware, applications, and network and support solutions quickly and at lower price points are growing at a rapid pace. IT will need to assemble services and solutions from many providers integrating these together in the optimum manner to meet business needs.

As many IT organizations are under great pressure to provide consistent levels of service with decreasing budgets, they often look towards rationalizing the technical and non-technical components that they control such as hardware, software, people and contracts. However, this approach can focus too heavily on the “parts” and forget that the business cares about the services made up by these parts. Many IT organizations are unable to accurately describe the business facing services they provide, the true costs of these services or how they are being consumed. Therefore, decisions on where to apply tight funds and how to take cost out are often made at the component level without a picture of the overall business impact.

To this end, IT is undergoing a fundamental change in terms of its expectations from the business and how it manages and operates itself. This change requires a transformation from a traditional focus on IT components and outputs (e.g. produce a report, install a server, implement an application) to a focus on business outcomes and value (e.g. reduce customer order wait times, quickly onboard a new employee, enable collaboration between internal employees and external partners, etc.).

There is a growing recognition that managing IT by technology fiefdoms no longer allows for the quality and agility expected by the business. IT must organize and operate by services being delivered, not the technologies being employed. A Services Thinking approach is critical towards helping IT transition to a service delivery organization focused on providing value to its customers at an acceptable cost and risk.

The transformation from engineering silos to services suddenly exposes a lot of waste. Much of this has been built up over many years. Services that are not cost effective, solutions that are over-architected beyond what the business needs. Slow delivery processes. Too many layers in the IT organization (when you operate by technology someone has to pull it all together so multiple layers historically have been created to “integrate” things).

A Services Thinking approach is needed and it must be a Lean one. Businesses are under greater pressure more than ever to move quickly or die. New technologies and business models can quickly put a large company out of business. For many years IT ran with the mantra “we are the business”. Well guess what? A slow wasteful and costly IT delivery organization will no longer cut it in today’s world. Solutions can’t wait years and sometimes even months to deploy. IT must move fast. It must operate at high velocity. It must operate Lean.

New concepts and practices around Agile, DevOps, Lean IT and other approaches have been evolving in the industry to deal with this transformation. All of these offer great promise, but lack the Services Thinking approach that is so critical to success. It is painful to watch some of these evolve, dancing around the concept of services and service value in their own disparate ways without ever getting to the point. These practices hold great promise. The tenet of this book is that they are all tremendously useful, but they need to be integrated into an overall approach that delivers high quality reliable services when the business needs them. This book pulls all these disciplines together. They operate within the overall umbrella of ITSM. ITSM is the glue that holds them all together.

This book will directly address the activities, steps and approach for running IT through a program of IT Service Management operating at high velocity. Much attention is spent on where waste typically exists within an IT delivery organization. It then turns to what you can do about it. An overall ITSM High Velocity Operating Model is presented to provide a context for how IT operates in this new world. Critical practices such as Agile, lean IT, DevOps and other evolving operating strategies are presented in terms of their IT Service Management impact. Included are a set of tools that you can employ that can assist you in finding waste, operating lean and improving your own IT delivery execution.

Why This Book Was Written

If you read through this book and still don't believe there is a critical need for IT Service Management then good luck seeing if you can survive in IT for the next 5 years.

There is a lot of information out there in areas such Agile, DevOps, Lean IT and other disciplines. Much of it is presented disparately. Much of it is driven by IT development expertise which woefully leaves out a lot of operational needs and concerns. This book intends to overcome that gap. IT Service Management (ITSM) is used as the glue for pulling these disparate disciplines together. The all operate under an overall ITSM framework because it can't be said enough:

"There is NO value in IT to the business until the point that a service is actually delivered"

End of story. The business could care less about whether development is now moving to Agile, whether the latest technology has been implemented, or whether IT has hired the smartest people. If the service is down, IT hasn't done its job. If development delivers the code faster and on time, but it can't be operated day-to-day at acceptable cost and risk, then IT hasn't done its job.

Not to say that Agile, DevOps, Lean IT and other approaches are not useful – they are definitely important. But they need to be utilized and integrated into an overall approach that delivers high quality reliable services when the business needs them.

This book will pull all these disciplines together. They operate within the overall umbrella of ITSM. The ITSM High Velocity Operating Model describes how. Careful attention has paid to concepts like Agile and Lean IT and apply them to real world IT examples.

It is hoped that the content in this book can serve as a reference guide to IT workers, be they developers, IT support staff, executives, middle management or project leads who are working on IT initiatives to help make them successful.

The Business Case For High Velocity ITSM

At the executive level, focus should be placed on outcomes not the means to those outcomes. If executives agree to those outcomes, then ITIL™ (IT Infrastructure Library) and ITSM merely become the vehicles by which those outcomes will be achieved. The outcomes in most cases will center on cost initiatives and concerns, sometimes risk, very rarely quality by itself. Although people talk a good game about quality and certainly everyone nods their head about it, rarely are checks written for quality by itself unless it leads to lower operating costs or risks. Telling executives they need to implement ITSM or ITIL may get heads to nod, but rarely funds to be provided. Focus on business, cost and risk issues.

Another error IT frequently makes with executives is focusing on *IT total costs* versus *IT unit costs*. With this mistake, IT keeps focus on the total cost of computing (i.e. the IT expense last year was $400M) which management may view as too expensive totally missing the fact that the services and volumes that IT has to process may have grown because of business events. Worse yet, this focus keeps IT viewed as an overhead in the eyes of executives. It's quite possible (and highly likely) that ITSM can reduce unit costs, thereby allowing the business to do more at lower unit costs even though the total computing cost may be higher.

Selection of a return on investment strategy to take with executives will be critical. Below is a table that identifies several strategies that can be taken as you develop your business case:

ROI Strategy	**Outcome**	**Examples**
Cost Cutting	Eliminate and reduce IT costs for hard dollar savings	• Unplanned labor costs for time spent on incidents, fixing bad changes, dealing with bad releases, etc. • Unplanned expenses such as those incurred for capacity shortfalls, staffing shortages to meet demand • Services being provided whose costs exceed their value i.e. delivering email at

		$3,000 per employee per year • Assets that lie unused or underused such as unused storage, servers with low utilization, unused data center space or low virtualization of hardware • Bad outsourcing deals, wasteful licensing arrangements, inefficient sourcing decisions
Cost Avoidance	Implement improvements that will result in lower delivery costs in the future than those that would be incurred if not done	• IT unit costs to deliver service XYZ are $3,000 per employee for 1,000 employees at a total cost of $3M • Headcount will grow to 2,000 employees by next year increasing our total service cost to $6M • ITSM improvements will be used to lower our unit cost to $1,500 per employee – therefore the total service costs next year will be $3M for a savings of $3M

Cost Shifting	Shift costs from non-productive investments to productive ones	• 87% of IS budget is funded for maintenance and support activities – if we improve our process efficiencies, and eliminate low value services we can reduce that to 65% and shift those savings into new projects or offer better services
Risk Mitigation	Mitigate operating risks that could lead to fines, penalties, bad publicity, loss of business, or customers choosing other providers	• Use ITSM as a means to implement a service “shield” or “buffer” between the complexities of IT service delivery and what the customers see and experience • Use ITSM to meet regulatory requirements such as COBIT, HIPPA or SOX or to avoid penalties • Use ITSM to reduce outages or improve the customer experience to avoid loss of those customers who may seek to get their

		services elsewhere • Use of high velocity ITSM practices to deal with suppliers and IT vendors that may already be delivering their products and services via the Cloud with continual release and deployment methods
Piggyback	Find a major project or business initiative that is already funded that could use ITSM to guarantee success	• Use ITSM to support a major business acquisition • Cloud computing initiatives cannot succeed without ITSM • Data Center move or consolidations • Large application development effort such as a new ERP system where there is a desire to build and operate with Lean, Agile and DevOps approaches
Market Capture	Implement operating improvements to compete in	• Use ITSM to gain a competitive edge • Prove compliance to win a large contract or

	the market place	attract new customers – customers may specifically ask for ITSM on service bids and proposals • Use of high velocity ITSM methods to get products to marketplace much faster

Book Chapters in Brief

Brief descriptions of remaining book chapters are as follows:

Chapter 2 – The High Velocity ITSM Operating Model

This chapter presents an overview of High Velocity ITSM and its operating model. Provides a summary view of the model and introduces the core operating disciplines of Lean IT, DevOps, Agile, Integrated Application Lifecycle Management (ALM), Cloud Computing, Virtualization and Automation. It also covers some high velocity topics related to Service Transition such as Containerization and Declarative Configuration Management. It also stresses the importance of organizing as an IT service provider to truly gain high velocity ITSM objectives.

Chapter 3 – Lean IT Applied To ITSM

This chapter delves into using Lean concepts for finding ITSM waste. Presented here is use of the Eight Deadly Wastes framework cast into ITSM terms. Examples of waste typically found in ITSM systems is identified. Another example that highlights the concept of Cascading Waste is also discussed. An example of a software request fulfillment workflow is used to show how 4 key strategies for reducing waste can be applied to streamline that workflow.

Chapter 4 – Agile Principles Applied To ITSM

Use of agile practices pose interesting and effective changes in how solutions are built. This chapter covers agile concepts and strategies from the ITSM practitioner perspective. An agile operating model for ITSM is presented that illustrates how agile works for developing solutions. An example of how an infrastructure project can be done in agile as well as a maintenance and support effort is also discussed.

Chapter 5 – DevOps Applied To ITSM

This chapter covers DevOps practices from an ITSM perspective. Presented is an operational framework for working with development teams. This framework includes checklists for both development and operations staff for each ITSM lifecycle stage. It also presents a strategy for providing development staff with a catalog of operational services. Service offerings shown include some ideas for developer support as well as how operational services shown can integrate into the solutions developers are building.

Chapter 6 – Virtualization Applied To ITSM

This chapter provides an overview of virtualization technology. How it works, what it can be used for and what kinds of processing workloads work well or not so well with virtualized technologies. Included is a discussion on Change Management with virtualized devices and how this alters the risk profile when assessing change impact.

Chapter 7– Integrated ALM Applied To ITSM

This chapter briefly covers Integrated Application Lifecycle Management or ALM. Just as ITSM tooling suites synchronize and integrate many of the IT Service Management functions, ALM synchronizes and integrates many of the application development functions.

Chapter 8 – Cloud Computing Impact On ITSM

This chapter covers the basics of Cloud Computing. For ITSM, the Cloud represents a delivery channel for IT services. It is during Service Strategy that key decisions are made on whether to deliver services via the Cloud, which model of Cloud to use and who the Cloud providers will be. This chapter provides an overview of different types of Clouds and delivery models. It also addresses considerations for security and privacy.

Chapter 9 – Leaning Out Service Transition

This chapter directly hits where most of IT waste and inefficiency exist – the Service Transition processes. Here is where you will find ideas for how to deal with Service Transition in a world of rapid change. Covered are ways to handle Change Management in this world. Topics around Continuous Release, Testing and Deployment are included. Discussions of newer deployment mechanisms such as Declarative Configuration Management and Containerization are also presented.

Chapter 10 – Leaning Out Service Operation

This chapter runs through the ITSM processes of Incident, Request Fulfillment, Monitoring and Event Management to identify waste areas and solutions. Included is a program description for the 40-40-40 Problem Management Program to aggressively reduce incidents and unplanned labor costs.

Chapter 11 – Leaning Out Service Design

This chapter identifies cascading waste issues that are most prevalent with Service Design tasks. Presented are 6 strategies for helping IT identify their business services. Included are topics on where cascading waste is found, how Cloud site recovery is changing the IT Service Continuity landscape, considerations for high availability, supplier management and capacity sizing.

Chapter 12 – Leaning Out Service Strategy

This chapter presents topics on more effective management of the overall IT strategy. Presented here is a 5 step approach for developing an IT strategy aligned with the business. This approach has been used by a number of companies and is extremely effective in identifying service costs and seeing the impact of waste from a cost perspective.

Chapter 13 – Driving Out Waste With CSI

This chapter presents some strategies for operating as a Lean Service provider and establishing a Lean CSI Program. Describes are program approach, roles and responsibilities as well as governance. Included is a topic on how to break out of firefighting mode which is a common barrier to refocusing IT efforts towards operating Lean.

Chapter 14 – ITSM Lean Toolkit

This chapter provides a compendium of lean tools that can be used for analyzing, planning and identifying causes of waste. Each tool is described individually along with a step by step description of how it can be employed and what kind of situations the tool is best structure for.

ITSMLib Download Site

There are a number of tools that you may find helpful related to this book. These can be downloaded through a facility called ITSMLib™. This facility provides access to real world working documentation, templates, tools and examples for almost any ITSM project. The library is structured to easily find knowledge and can be easily searched with phrases and keywords to find relevant information. You may find this useful in jump starting your own solutions with ideas and content that has worked for others.

Here you can find many tools and aids that can help you jump start your ITSM efforts. Requests for access can be made to:

RandyASteinberg@gmail.com

When requesting access, please provide:

1) Your Name
2) An Email address
3) The company you work for

4) The country you reside in

This library is continually updated with content and we try to add things all the time as people see a need.

Chapter 2

The High Velocity ITSM Operating Model

High Velocity ITSM Overview

What is high velocity ITSM?

It is operating, governing and managing IT by service delivered with a keen focus on speed (velocity), efficiency, value, and cost. The focus on service glues together development and operations (utility and warranty) in a seamless solution. Continual Service Improvement is always in play, constantly looking for more efficient and effective ways of delivering defect-free services. Delivery is focused around service integration versus throwing technologies at the business.

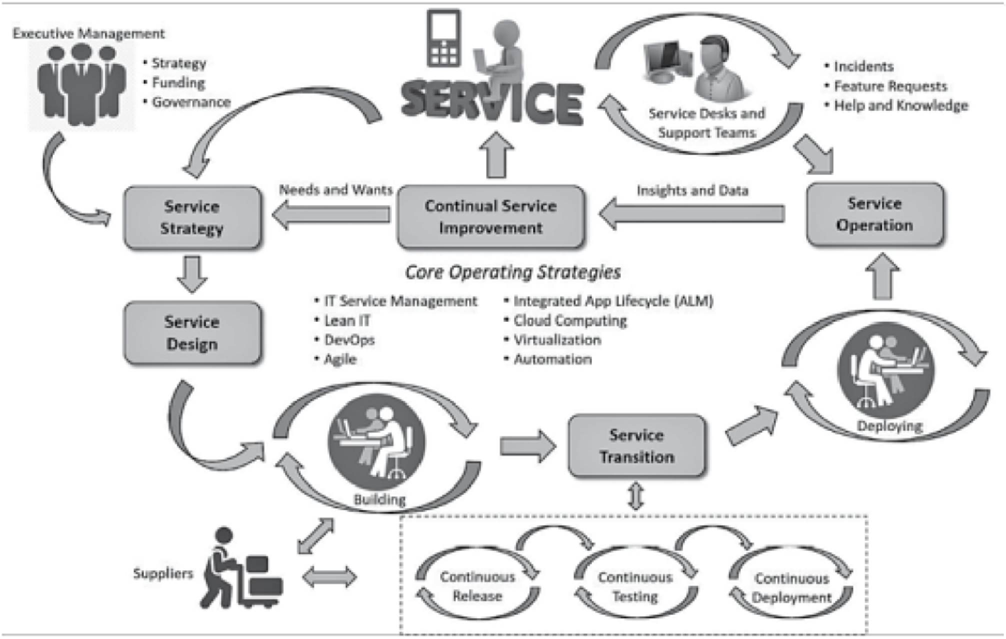
With high velocity ITSM, IT is always on the lookout for waste. There is a constant look and re-look at how services are managed and delivered. How processes operate to manage and deliver those services. Are the most efficient means in place? Is waste eliminated? Can opportunities with virtualization, automation and self-service be exploited? Can delays and wait times be eliminated? Which activities provide value versus not? How to best handle activities that don't add value but still must be done? In short, how can we best make IT operate faster, leaner, and transparently at acceptable levels of cost and risk?

High Velocity ITSM Operating Model

Presented here is a model for operating IT Service Management activities with high velocity. The model uses the ITSM Service

Lifecycle as a key foundation for how service solutions are strategized, designed, built, tested, deployed, operated and improved. With this model, the Service Lifecycle has been expanded to integrate modern core operating disciplines like DevOps, Lean IT and Agile. The lifecycle acts as the glue that holds those disciplines together.

The model is shown as follows:



Core Operating Strategies

Let's start with the middle, the Core Operating Strategies. These represent key disciplines that will be utilized as part of the high velocity ITSM system. A brief description is presented below. More detail is provided in the next chapters of this book.

IT Service Management (ITSM)

It all comes down to the service. The business sees no value in IT until the point a service is actually delivered. The disciplines of IT Service Management (Service Strategy, Service Design, Service

Transition, Service Operation and Continual Service Improvement) are more important than ever. ITSM is the glue that holds everything together. It runs from strategy to delivery and as can be seen above, forms the main flow of the entire lifecycle.

Lean IT

This is an extension of lean manufacturing and service principles to the development and management of IT products and services. The objective is to aggressively look at all the IT management processes and services and aggressively look for waste. There is a constant lookout for opportunities to become more efficient. This can include areas such as eliminating slow or inefficient process steps, automating common redundant tasks, increasing self-service, leveraging technologies such as virtualization that utilize physical resources more efficiently or Cloud Computing to eliminate the need to have to install and manage physical resources on business premises.

DevOps

A culture and practice that emphasizes the collaboration of software developers and IT Operations to allow for rapid and frequent build, test, and release of reliable software. Gone are the practices of throwing development solutions over the wall to IT operations. These forces now work in tandem at all stages of the development lifecycle to release solutions together. To deal with rapid change, continuous release, testing and deployment are in place. The objective is control changes and releases into the production environment, but do it much more efficiently such that the business does not get unnecessarily slowed down with long lead times to get new or changed solutions.

Agile

This is a solution development practice in which requirements and solutions evolve and continuously improve with rapid and flexible response to change. Agile practices focus on releasing fast with small increments of change versus large big-bang approaches. Rather than incur long wait times and asking the business to sign off on big-bang solutions, users get to see results much faster and provide continuous feedback that can be quickly incorporated into each agile release. Software itself is maintained in an agile manner, such that the latest release of software is always in production. This avoids lengthy projects and delays related to software upgrade efforts. Software defects are fixed much faster.

Integrated Application Lifecycle Management (ALM)

Taking a page from the ITSM community, the development community has recognized that you cannot continue to build solutions in engineering silos with myriads of disparate development and management tools that don't talk to each other. Activities such as version control, bug tracking, code management, project management, requirements tracking have historically been handled by individual toolsets. Tooling vendors have been aggressively looking at Agile and DevOps practices and are now coming out with integrated suites of development related tools that integrate development activities and information across the many development tasks.

Cloud Computing

This is about use of Internet-based hosting services to provide on-demand access to a shared pool of computing resources (e.g., networks, servers, storage, platforms, applications and services) which can be rapidly provisioned to achieve economies of scale. Efficiencies are created by having a 3rd party handle all the complex infrastructure management of physical resources, securing them and making sure they are available in the event of a disaster. Provisioning activities are done within minutes versus the typical 2-3 month timeframe to approve, fund, obtain, install and configure resources onsite.

Virtualization

Virtualization is about separating operating systems, data and platforms from the hardware they run on. Technologies have advanced quite a bit within the last several years to virtualize almost everything IT does that runs on physical resources. Servers, desktops, networks, storage and even data centers themselves can now be virtualized. While each choice taken here needs to be considered in terms of cost and risk, the advancements in this area should be continually examined. In today's world, it is not uncommon to see large amounts of servers, storage and desktops virtualized in many IT organizations.

Automation

IT automation is the handling of manual tasks through scripts and software in such a way that those tasks become self-acting. The benefits of automation are huge. It reduces or eliminates the need for labor, it responds much faster than people, it executes on tasks in a consistent and controlled manner and prevents errors and defects from occurring related to human mistakes. While ITSM may be the underlying foundation for high velocity delivery, it is automation that links everything together and makes it flow. Almost every task in IT can be looked at with an eye towards automation.

Included in the model is an expansion of the ITSM Service Transition stage. Concepts around Continuous Release, Testing and Deployment are used to vastly speed up the pace of change in IT services and solutions. More details can be found in the Leaning Out Service Transition chapter of this book. In addition, several key evolving industry practices are described that play a key role in speeding up Service Transition activities. These are:

Containerization

Initially used for effective and efficient application deployments, the High Velocity ITSM Model introduces the concept of deployment by "containers" for all infrastructure components that support services. Use of this executes deployments and request fulfillment activities faster, avoids human errors and simplifies Capacity Management activities.

Declarative Configuration Management

This involves taking advantage of new technologies emerging in the industry to auto-deploy Configuration Items. Rather than treat the practice of Configuration Management as a static provider of

Configuration Information, this goes one step further and makes that information actionable. If all the attributes of a Configuration Item are already known, then why not pump those attributes into an automatic script that auto-deploys that Configuration Item?

The Model Lifecycle

Let’s walk through the model at a high level. We will start at the upper left of the diagram and move counter-clockwise through each of the key elements.

Executive Management

It is here where key service decisions and strategies are made. What services should IT offer? How much do we invest and in which services? When should we invest? Are the services we offer in line with business goals and objectives? Are services delivering value? What kinds of projects need to be put into place to create, modify or remove the services being offered? Activities are strongly linked with Service Strategy. That strategy is represented and governed through an IT Service Portfolio.

Service Strategy

It is in this lifecycle step where the IT Service Portfolio is governed and managed. Executive decisions are made to determine where to invest, what projects to undertake or how IT will organize and operate to build and deliver services to the business. The outputs of this stage lead to projects that will build new services, modify or enhance existing services or remove services no longer needed.

Service Design

In this lifecycle step, activities take place to construct designs and specifications for building IT services. With a high velocity approach there is one key change here. Non-agile approaches might use this stage to build detailed designs that include every requirement needed by the business. With an agile approach, build activities proceed without knowing every requirement that may be ultimately needed. The business is not being asked to sign off on every requirement before build work can start. Therefore, this stage focuses at a higher level on things like Capabilities (What the service needs to provide), User Stories (discrete business needs/benefits) and service features that will be developed. These are then entered into ALM (Application Lifecycle Management) tools to kick off development team efforts. Larger efforts may include Release Trains which are combinations of multiple build teams that work together and release together.

Building

Solution building takes place here. DevOps practices are used to get development and operations to work together to develop comprehensive service solutions. Development teams focus on service utility – what features the service needs to have. Operations teams focus on service warranty – putting the infrastructure in place to make sure the service will be there when needed and will operate day-to-day at acceptable cost and risk. Suppliers are brought in to integrate their services with the solution being constructed. Build activities take place with agile disciplines. Solutions are released through sprints which produce production ready services in small increments of function on a frequent and regularly scheduled basis.

Service Transition

Here is where activities take place to handle testing, change, release and deployment tasks. While those activities are needed to manage and control what goes into the live production environment, there is one big change here. That is that those items now operate in continuous manner. Releases are always in play and managed through ALM tools that synchronize development code and branches. Testing operates continuously wherein new releases get thrown into the testing mix and thrown out if defects are found. Release to the live environment or staging area occurs at any time whenever testing activities have run a cycle without defects.

Deployment

Actions now take place to transfer solutions to user devices, workstations, system or other production environments. The mechanics of deployment leverage automation. Containerization and Declarative Configuration Management are used to automate and speed up deployment activities. Not all deployment will be immediate. There may be good business reasons to make sure that deployment does not occur during heavy business periods of activity (like a yearend closing) or that end users need to be trained. Sometimes logistics require that solution deployments be throttled back such as deployment of code to regions of the world that have network bandwidth issues.

Service Operation

In this lifecycle stage, solutions are operated day-to-day. Typical activities include handling incidents and problems, fulfilling requests, monitoring the operating state of solution components and performing administration and support tasks. Two elements make this one of the more important stages in the lifecycle. It is at this point that the business actually receives value from their IT investment. It also generates lots of insights and data around service quality, performance and business complaints.

Service Desks and Support Teams

These entities serve to provide contact points for the business to interact with IT to fix defects, fulfill requests and get help. Since the Service Desk is one of the first and main contacts between IT and the business, it provides a valuable resource for information around service defects, how services are being used and whether the business is truly finding value in the services they are receiving.

Continual Service Improvement

With this lifecycle stage, issues and defects are captured and analyzed for improvement opportunities. Insights are gained on how to execute services and manage supporting processes more effectively. Those opportunities are not just for application defects. A continual effort is made to reexamine how services are being delivered looking for opportunities to deliver faster and at lower cost. A Continual Improvement Register is kept to list these opportunities. This is shared across IT and with executives who may use that information to tweak and tune their strategies.

Organizing As An IT Service Provider

You can’t solve the issues of higher operating velocity purely through newer technologies and processes. IT also needs to organize as a service provider.

IT cannot manage itself as it has in the past primarily by technology silos in a world where services depend on a well-coordinated chain of delivery technologies individually managed by those silos. The service is the sum of what is delivered from all the technology silos that support it. If one silo fails, the service fails. Therefore, accountability needs to be built into the organization for the overall service.

Without this, it is IT executive leadership that is stuck providing the coordination and integration points to get services to work. This is a situation that IT leaders constantly complain about. Why can’t IT get its act together? Why do they have to be on the phone over and over to many IT support teams just to get things done?

Even worse, what happens if IT leadership itself abandons the role of service integration? The result is not pretty. At that point, the business ends up performing coordination and integration activities just to make sure their services will deliver. Their view of IT will be very low (“they can never get anything done” they will say) and satisfaction with IT services will be at a horrible level.

Therefore, the need for service integration is always there whether IT wants it or not. If IT operates in a way that is inconsistent with service management, then those activities will drift upward towards the business. If the business feels they need to take over those activities, they will go out and buy their own tools and services outside of the IT organization.

Operation of services delivered at high velocity must include the structure of an IT organization that is focused and accountable for services. Organizing around technologies and capabilities is not enough. Imagine running an Agile Scrum session and there is no Product Owner? Imagine operating in the role of CIO and there is no one to turn to when a key service goes down?

Accountability for each service needs to be in place. IT career paths need to diverge allowing for technology specialists who understand how things work on an equal footing with those who understand how to bundle and integrate disparate technologies and service offerings from others into a seamless service that customers can understand and consume.

The role of Service Owner is the most important organizational change needed to be successful. With this role, a Service Owner is accountable end-to-end for delivery of the service. If the service experiences an incident, it doesn't matter where the root cause lies or which team needs to be called in. The service is still out and it is a Service Owner responsibility to get the service back up, even if that includes coordinating with many teams to make that happen. No more pushing tickets back to the Service Desk!

The Service Owner role should be established at an organization level that allows them to interoperate across existing support teams and technology silos. It is suggested that this role be placed within a new organization silo that reports directly to IT executive leadership. As an alternative, this role could exist within a logical organization structure drawn from the existing silos. In this case, the logical organization has a direct line to the IT executive leadership.

The Service Owner role not only provides a single point of contact function for a service, but also works to improve service delivery performance. Broad-based measures that cut across technology silos are used to gauge service performance. The failing of one or more technology silos no matter where they exist will be reflected in the incident counts, problem counts, service breaches and delivery measures for the service as a whole.

The Service Owner should be accountable for how well the service performs. Include in their job description the responsibility accountability for taking action when service performance is out of spec or trending out of spec. Give them organizational power. For example, they should have access to senior leaders, involvement in all major and monthly review meetings, budget control, a title, and a say in rewards and incentives for line managers.

Secondary to the Service Owner role is the role of Business Relationship Manager (BRM). This role provides the voice of the customer view on service delivery activities, customer requirements, needs and wants. It is critical to communicating with the business. BRMs can be either in IT or in the business units. They are accountable for IT meeting business unit needs.

Organizing as an IT service provider is discussed at length in the Organizing ITSM book. In that book you will find recommended roles and responsibilities and examples of organization structures for different delivery models.

The key takeaway from this book should be the concept that organizing as a service provider is key and ensure that an acceptable organizational solution is considered to be truly high velocity. This cannot be covered over with different processes, tools or use of leading agile practices.

Chapter 3

Lean IT Applied To ITSM

Lean IT Overview

Applying Lean concepts to ITSM is about eliminating waste throughout the IT organization. Waste results in higher operating costs due to unplanned labor activities, rework, and delays. It takes IT labor focus away from projects that will move the business forward and ties it down to reactive fix and repair activities. Lean concepts have been around for years, primarily in the manufacturing industry. The objective of this chapter is to take those same concepts and apply them to IT. By doing this, IT resources and labor can focus on activities that truly help end users, impact the business, and grow the bottom line.

Lean IT promises to identify and eradicate waste that otherwise contributes to poor customer service, lost business, higher than necessary business costs, and lost employee productivity. To these ends, it targets eight elements within the IT enterprise that add no value to the finished product or service or to the parent organization.

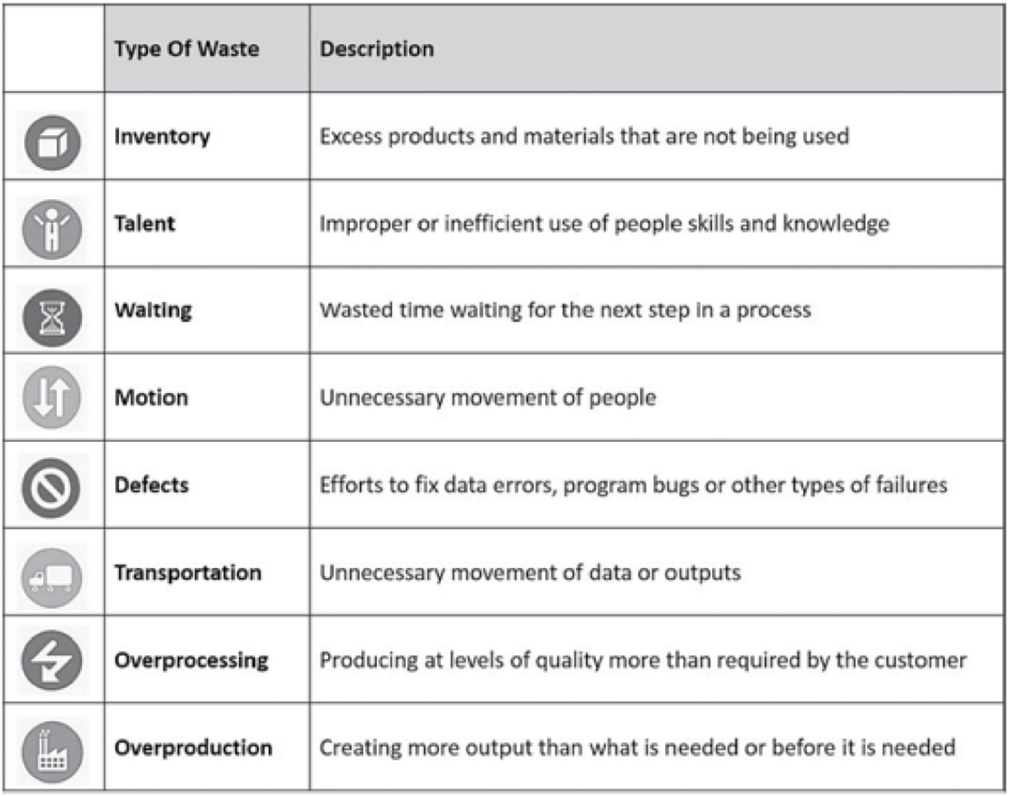
To accomplish Lean IT objectives, the Eight Deadly Wastes tool (See Lean Toolkit chapter for more detail) will be used to provide a general framework for looking at waste within a Service Management system. The framework will be used to identify where IT waste is typically found and what opportunities might exist to remove that waste.

This means applying each of the waste categories from the Eight Deadly Wastes tool to the services, the IT organization is delivering. Each service, service lifecycle process, application, request and operational workflow, infrastructure area and development component is assessed against each of the eight waste categories.

For each category, this chapter provides a small list of typical IT ways of operating that might be starting points for finding waste. A list is presented for each waste category. While nowhere complete, it is hoped that the items listed provide enough hints to help you find waste within your own IT organization.

A Framework For Finding Waste

The framework is shown below:



	Type Of Waste	Description
	Inventory	Excess products and materials that are not being used
	Talent	Improper or inefficient use of people skills and knowledge
	Waiting	Wasted time waiting for the next step in a process
	Motion	Unnecessary movement of people
	Defects	Efforts to fix data errors, program bugs or other types of failures
	Transportation	Unnecessary movement of data or outputs
	Overprocessing	Producing at levels of quality more than required by the customer
	Overproduction	Creating more output than what is needed or before it is needed

The types of waste listed in the table above are pretty typical of most Lean programs found in manufacturing and other industries. They also easily apply to IT. Let’s look at each one individually and see how these kinds of wastes are typically found throughout almost any IT organization.

Waste Category #1 - Inventory

This refers to excess product and materials for which labor or other resources is being expended yet those items are not being used. ITSM related examples of this kind of waste can include:

- Production of system documentation that is never used by IT support staff
- System resources that are running at low utilizations (e.g. storage pools where only 5% of that resource is actually used, servers and network lines that are running at utilization rates below 15%)
- Software license agreements where large numbers of licenses have been purchased but only a small number are being used
- Application software that is being built or purchased, yet not really used
- Data Center facilities where less than half the floor space is used
- Development teams that are sitting idle
- Services being provided and managed that no one really cares about
- Poor scheduling and leveling of IT resources to meet service demand volumes
- Staff work areas that are half-empty and under-utilized.

Keep an eye out for anything tangible that appears to not be in use or for items being produced that no one uses. Focus on sizeable opportunities. Inventory waste is one of the easiest things to fix and can provide significant early wins in your efforts. Some challenges might exist with individuals who may want to hold onto their pet projects and resources. Making the cost savings public is a good way to overcome those challenges.

Waste Category #2 - Talent

This refers to a misuse of people and skills. ITSM related examples of this kind of waste can include:

- Spending a majority of staff labor on reactive and unplanned non-value tasks (e.g. fixing incidents, bad changes, rework) versus proactive improvement projects or building new services
- Spending large amounts of time researching issues and obtaining information because knowledge wasn’t readily available
- Staff resources assigned to projects that no one cares about
- Use of expensive resources doing common IT tasks that lower cost labor resources could handle (e.g. too many requests or incidents are being escalated to Level 2 staff when Level 1 staff could easily handle given proper training or knowledge)
- Not taking advantage of the full set of skills and experience of existing support or development staff (e.g. may have an individual with 2 Masters degrees working as a Service Desk analyst)
- Utilizing development and support resources that are under skilled for the tasks they need to undertake
- Poor retention of skilled resources
- Staff resources spend inordinate amounts of time on mundane or repetitive tasks that could be better automated
- Failing to capture good ideas and innovation from support and development staff
- Major incident process that requires all support resources to be on a bridge call when in reality only 2 people are doing most of the talking
- Meetings that take place week after week that provide little value yet consume lots of resource time
- Use of less skilled resources on tasks that require highly skilled

resources that could result in those tasks being done much more slowly or with errors

- Failing to obtain support staff points of view on how issues and problems might be best addressed or where they may see efficiencies that could be implemented.

With this category, examine how staffing resources are being used. Do they have additional talents and skills that could be employed on other initiatives that could provide value? Are they being challenged to provide ideas and solutions that would allow IT to operate more efficiently and at lower costs and risks? Are expensive resources being used where lower cost resources may be able to provide similar outcomes? Is too much time being diverted to unplanned work away from projects and initiatives that could provide more value?

Waste Category #3 - Waiting

This category refers to lead time where a process is simply wasting time waiting to get to the next step in the process. ITSM related examples of this kind of waste can include:

- Change management process where 80%+ of the time is spent waiting for approvals to happen
- Software provisioning process where days or weeks are spent waiting for approvals to get the software
- Scheduling constraints that push back target dates and deadlines because reviews, implementation and fulfillment tasks couldn’t be scheduled in line with target outcome timeframes
- Server provisioning times that take weeks or even months waiting for a technical support resource to get assigned
- Large scale deployments where sites may have to wait months or even years to get a new or upgraded solution
- Waiting to reach a Service Desk representative to handle a situation which could easily be done via a self-service facility
- Holding back on development progress until every possible business requirement is identified and waiting until the business

signs off on all the requirements

- Finger pointing across the IT organization as to who is responsible for support, provisioning or request fulfillment activities
- Holding off on release of a new application or service until every possible feature is put into the release
- Holding off on a release of a new application or service until it passes a lengthy battery of test runs and quality assurance reviews
- Fulfilling requests in serial fashion versus batching up requests to possibly handle them all simultaneously
- Long queue times that are the result of not enough resources to handle the workload demand placed on them
- Delays caused by escalation of support tickets (incidents and problems) to other support staff
- Delays caused by waiting for too many decisions to be made (e.g. staff not empowered to make decisions, therefore management becomes a bottleneck to get timely decisions made).

With this category, hone in on places where unnecessary delay time occurs waiting for something. Examine each place where a queue, signoff, approval, or decision is required. Are these really necessary? Can work be batched to handle multiple things at the same time or must each task wait until the next one finishes? If wait times need to occur, can they be minimized? If large scale deployments are taking place, can these be done more simultaneously versus one site after the other?

Waste Category #4 - Motion

This category refers to unnecessary movement of people. ITSM related examples of this kind of waste can include:

- Dispatching support services out to local sites when the work could be done remotely

- Utilizing support staff to handle tasks that could be automated
- Flying staff or managers out to locations to attend meetings when those meetings could be held virtually without the need for travel
- Locating IT staff resources physically far from their management or others they need to frequently communicate with such that staff need to travel from one end of the campus to another to hold frequent meetings
- Excessive efforts spent to search for information such as configuration settings, knowledge, hardware and software inventories
- Chasing down approvals where multiple parties need to be involved in the final decision
- Re-solving incidents over and over because diagnostic and work-around solutions have not been published or communicated
- Chasing down approvals that are not really necessary
- Locating staff resources in facilities are locations that are far from basic resources such as food, parking or bathrooms that consume large amounts of time to get to
- Not providing enough meeting space or conference rooms such that meetings and activities get delayed until those facilities become available (e.g. "we can't meet at 9am as no rooms are available until after 2pm…").

With this category, look at people in the IT organization and how they physically move around. Are large chunks of time spent going out to lunch because there is no cafeteria nearby? Are people constantly traveling from one end of the building or campus to the other because they are in disparate locations? Is work being done onsite that could be done remotely? Unnecessary motion can really chew up time. Be on the watch not only for those situations that require long travel, but also for small amounts of time that are wasted in this category that occur frequently. It may take only 5 minutes to walk down to the manager's office, but if this happens ten times a day that adds up to almost an hour a day, 5 hours a week, 20 hours a month or 240 hours a year. That's almost 6 weeks out a year wasted!

Waste Category #5 - Defects

Perhaps no other category wastes more time and resources in IT than this one. Defects result in unplanned labor to address. They impact not only the flow of work in the IT organization, but also flows of work within the business itself. If serious enough, they erode business confidence in IT capabilities and can even impact business customers and the reputation of the business itself. ITSM related examples of this kind of waste can include:

- Incidents and problems that disrupt applications, infrastructure and services
- Failed changes that disrupt production
- Releases that contain program errors and application bugs
- Solutions that may work, but still fail to meet customer requirements
- Unfilled service expectations caused by lack of awareness of what those expectations should be
- Slow application response times that frustrate business users
- Inaccurate data that is being produced
- Defects caused by unauthorized system and application changes
- Defects caused by poor project execution
- Defects caused by poor product designs
- Development of applications and services that are not fit for use by the business
- Defects caused by services that cannot be operated day-to-day at acceptable cost and risk
- Failed deployments
- Services and applications put into production that introduce security risks
- Delivering solutions that are not meeting business needs
- Improperly deploying services into production that are not

operationally ready

- Defects that are discovered late in the solution development or transition cycle such that they are much more expensive to fix than if they had been found earlier.

With this category, look around IT for where defects occur. Incident tickets make a great start. Categorize these (if not properly categorized already) to hone in on areas that might be addressed. Unplanned labor is also another good metric to look at. Another is what happens when a new release goes live. If introduction of new applications and services causes higher rates of incidents and support confusion, then opportunities may exist to remove this waste through a better release or operational readiness process.

What percent of the time do people spend fixing and repairing incidents and issues versus working on projects and improvements? If that time is significant, eliminating defects may carry a lot of value.

Waste Category #6 - Transportation

This category covers waste found in unnecessary movement of data or outputs from a process. Many times in IT, data is needlessly moved around from one system to another. Forms that are filled out manually might be rekeyed into an IT system. ITSM related examples of this kind of waste can include:

- IT systems where data is created in one system and then moved to another system in order to support a service or business function
- Asking users to manually fill out a form and then manually rekeying the filled out form into an IT system
- Deploying desktop images from a central source to a large number of desktops located at remote sites versus deploying to a much smaller number of image servers located at sites where those desktops exist and then letting desktops pull from their local server

- Deploying entire images versus just the applications and components that have actually changed
- Having to physically observe, count and audit hardware and software for compliance with asset and security policies
- Having end users ship their desktops into a central depot facility in order to get them repaired and then having to wait until the repaired desktop is shipped back to them (why not just ship a spare desktop as soon the user reports an issue?)
- Allowing users to store data on personal hard drives which then takes hours, days or even weeks to recover when they get a new or repaired desktop
- Systems that produce CSV (Comma, Separated Value) files or other output formats unnecessarily in order to feed that output into another system
- Managing pools of physical servers which have to be manually commissioned and decommissioned versus leveraging use of virtual servers
- Applications that create temporary work files on inefficient media such as tape, disk, or CD versus using memory or electronic storage
- Managing, configuring and deploying physical desktops versus leveraging virtualized desktop technologies
- Running sub-optimized network configurations that make movement of data occur over too many hops and routes
- Needlessly batching outputs when these could be input directly to the next system in the workflow
- Mailing or even emailing reports to end users when these might be better viewed in a shared portal or dashboard.

Similar to the Motion Category, examine each place where data is being moved from one system to the next. Is this really necessary? Are forms and data being reformatted or rekeyed in order to move from one system to the next? Could the production of data from one system directly flow into another system?

Waste Category #7 - Overprocessing

This category addresses processes and tasks that spend resources, time and effort delivering far above what is needed. ITSM related examples of this kind of waste can include:

- Applications that are constructed to deliver many more features than what the business has asked for
- Excessive documentation and review procedures
- Using forms and screens that slow down users because they force formats that are unnecessary or could be determined in the background (e.g. forms where the user has to enter their phone number one number at a time into a series of formatted boxes – why not just give them a blank box to enter their number?)
- Buying and using software that is overkill for the functions that were originally required (e.g. purchased an expensive configuration management database system just to keep inventories of hardware and software)
- Operating with multiple tools that all do similar functions
- Forms and screens that ask for much more information than what is needed to accomplish their mission (e.g. does a password reset request really need 3 pages of detail filled out?)
- Using multiple tools to provide the best of breed across every function needed versus a single tool that provides the basic needs across all those functions (e.g. use of multiple monitoring tools because different products are best of breed on different operating systems and networks)
- Tracking assets and configurations down the minutest detail when no one cares or it is not cost effective to do so (e.g. tracking PC cords, cables and mice into an asset tracking system which adds considerable administrative overhead when in reality, these items can be cheaply replaced)
- Delivering IT services that are seldom used or no one really cares about
- Solutions that are crafted to include and handle every possible

exception that might occur

- Implementing custom solutions when off-the-shelf solutions already deliver most of what is needed
- Implementing a stringent change management process and set of controls to handle low risk/low impact or standard changes
- Running a long set of comprehensive and stringent applications test cycles on an application that has low risk/low impact to the business organization
- Delivering IT services that cost much more to operate and deliver than the value that they actually provide.

IT historically goes for perfection. Many times there is the thinking that unless every business requirement, function or feature is implemented the solution will not be acceptable. It is easy to over-architect solutions and build much more than what the business would be happy with. Constructing more than what is really needed is a form of waste. It is not unusual to find millions of dollars in IT solutions that are over-architected, expensive to operate, and cost much more than the value they provide to the business. Examine each solution delivered by your IT organization to look for potential opportunities in this category.

Waste Category #8 - Overproduction

While the Overprocessing Category focuses on systems and services that have been implemented with more functions and features than necessary, this category focuses on whether the outputs from applications and services produce more than necessary. ITSM related examples of this kind of waste can include:

- Operating with more network, storage, memory or processing capacity than what is really needed or actually used
- Overproduction of outputs related to poor customer communications or definition of requirements
- Production of reports and dashboards that no one uses or looks at

- Screens and forms that are seldom used
- Delivery of low value applications and services
- Reports and dashboards that include many more metrics and information than what people need or see value in using
- Reports, dashboards, service catalogs and portals that provide much more detail and information than what is really needed
- Implementing software patches and updates that may not be needed
- Producing system documentation, user guides or other form of documentation that provide much more than the recipients of those items needs to know
- Providing too many emails and communications about the status of user requests
- Producing monitoring and event data with details far beyond what most support staff need or look at
- Services that being delivered at higher service targets than what was originally agreed to through SLAs or other agreements
- Level 2 support staff that responds directly to end user calls to resolve issues or fulfill requests that never go through the Service Desk
- Collecting and recording log data at intervals that are much more discrete than what is needed (e.g. logging server activity every second versus once every 2 minutes)
- Providing state-of-the-art customer care at the Service Desk when only basic call handling and support functions are needed.

Similar to the Overprocessing Category, examine each service, system, application or infrastructure that is in place to see if it is delivering at levels needed by the business. Delivering more than what is needed is not necessarily a good thing. Those extra items can cost more in resource and labor, set the wrong expectations with users and take resources away from more valuable projects and improvements.

Cascading Waste

Don't just look at waste categories previously described by themselves. Sometimes combinations of those categories can create even more significant amounts of waste. A domino effect can occur where the occurrence of one form of waste in one category can cause another form of waste in another category. This is known as cascading waste. Consider the following example:

- Defective server hardware (Defects) may cause a major incident service outage that in turn causes a lengthy wait for users of that service (Waiting)
- This results in excessive demand on the Service Desk (Motion)
- The excessive demand elongates the call and dispatch queues (Waiting)
- More Service Desk agents are brought in to deal with the demand (Inventory)
- The impact is such that senior management needs to get involved (Motion)
- Skilled resources need to drop projects they are working on to address the incidents (Talent)
- The business now uses manual work arounds to deal with their customers (Motion)
- IT Business Relationship Managers are dispatched to business locations to quell concerns about service availability (Transportation)
- Server architectures are updated with duplicate servers and failover capabilities to avoid the incident in the future (Overprocessing and Inventory)

Perhaps upgrading server hardware in the first place might not be quite so expensive when you consider the impact of cascading waste!

Leaning Out Processes

Much has been said in this book about how to identify waste in an ITSM system. This chapter now addresses some strategies for dealing with it.

- **Focus On Each Process** – what steps provide value? What steps introduce delays? (e.g. 90% of change management time is spent waiting for an approval, users wanting software wait days or weeks to get approvals – why not give it to them right away and the revoke later if not authorized?

- **Automate Common Tasks** – institute an aggressive automation program to automate common fulfillment tasks and/or reduce human engagement

- **Increase Self-Service** – extend portals and the IT Service Catalog to allow users to engage more and more with IT self-service

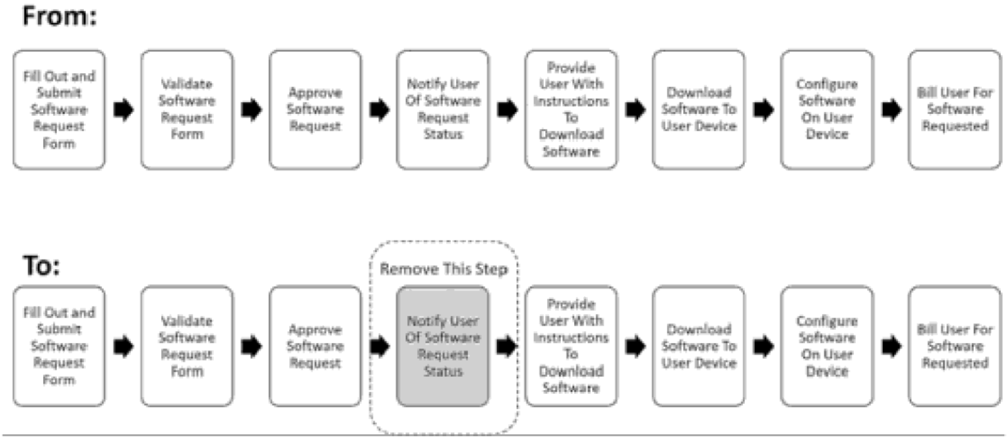
- **Leverage Virtualization** – virtualize servers and PCs to speed up provisioning activities and provide more efficient management and support

When looking at a process to make it more efficient, here are some key strategies you might employ:

Strategy #1 – Eliminate Unneeded Process Steps

With this approach, lay out each process in terms of its steps, identify which steps provide value and which do not. Then remove those steps that do not provide value. The Lean Toolkit presented at the end of this book has a number of tools such as Value Stream Analysis or Non-Value Step Analysis that can help.

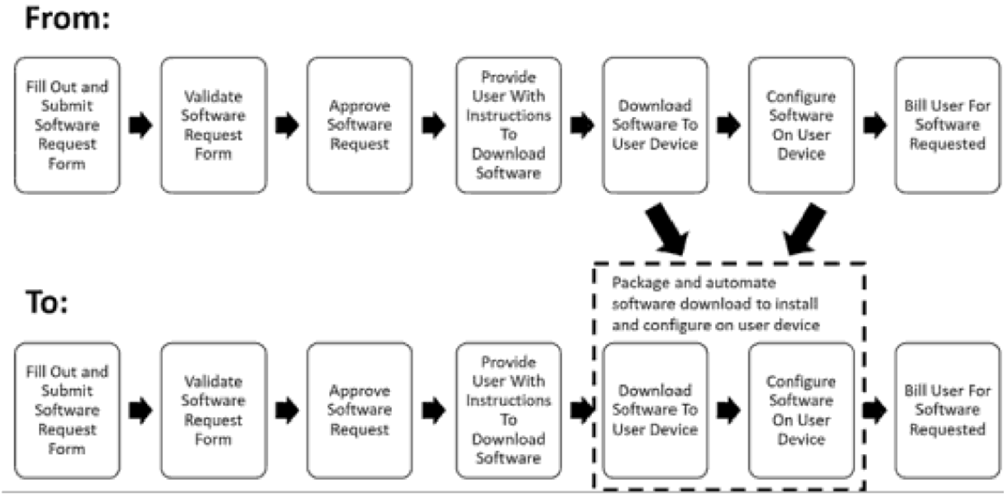
Example:


From:
Fill Out and Submit Software Request Form
Validate Software Request Form
Approve Software Request
Notify User Of Software Request Status
Provide User With Instructions To Download Software
Download Software To User Device
Configure Software On User Device
Bill User For Software Requested
To:
Remove This Step
Fill Out and Submit Software Request Form
Validate Software Request Form
Approve Software Request
Notify User Of Software Request Status
Provide User With Instructions To Download Software
Download Software To User Device
Configure Software On User Device
Bill User For Software Requested

With the above, the "Notify User Of Software Request Status" might be considered an extra step as the notification is already given when the user instructions are given in the next step.

Strategy #2 – Combine Or Consolidate Steps

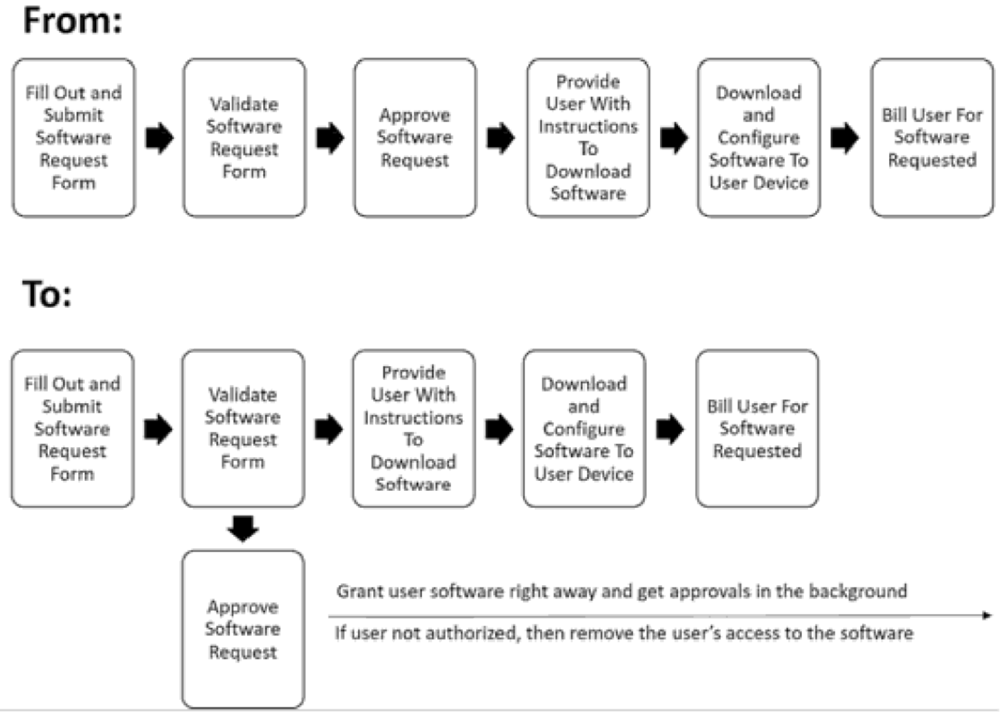
For this strategy, some steps might be able to be combined into a single step.


From:
Fill Out and Submit Software Request Form
Validate Software Request Form
Approve Software Request
Provide User With Instructions To Download Software
Download Software To User Device
Configure Software On User Device
Bill User For Software Requested
To:
Package and automate software download to install and configure on user device
Fill Out and Submit Software Request Form
Validate Software Request Form
Approve Software Request
Provide User With Instructions To Download Software
Download Software To User Device
Configure Software On User Device
Bill User For Software Requested

Here it has been determined that activities to download software can be automated such that the download and configuration steps can be done in a single step.

Strategy #3 – Move Steps That Cause Delays Into The Background

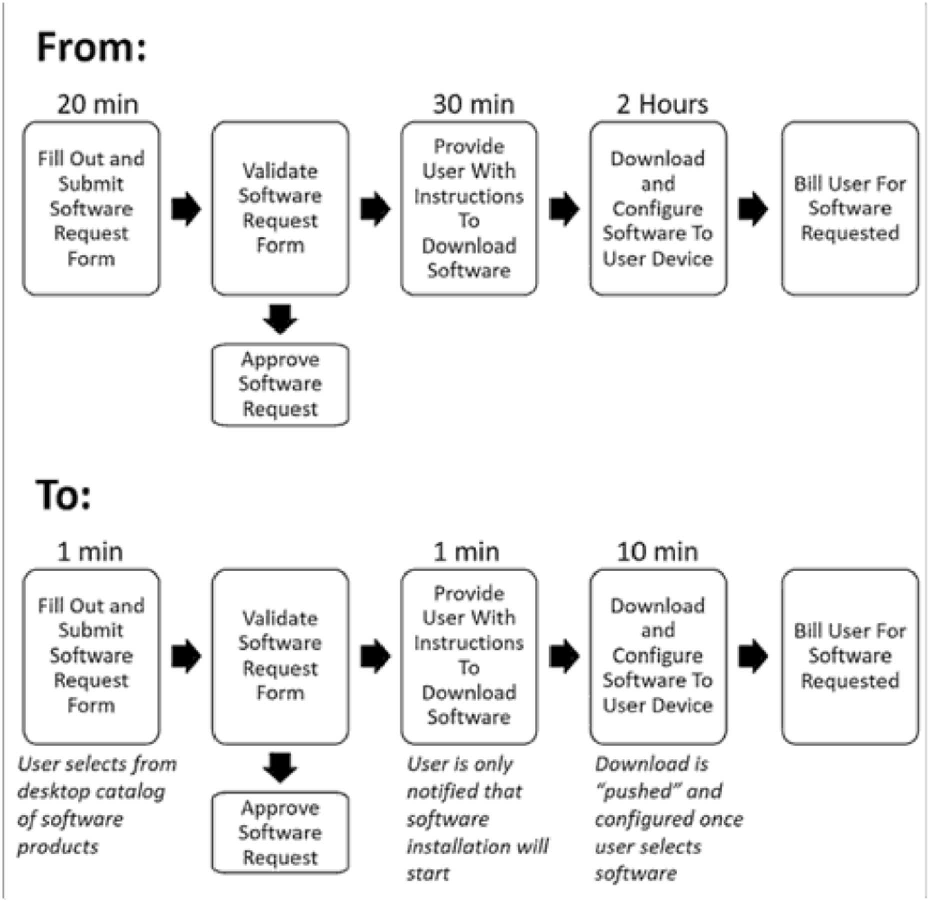
For this strategy, take steps that are non-value and cause delays and move them into the background such that they execute in parallel and have removed delays from the user.


From:
Fill Out and Submit Software Request Form
Validate Software Request Form
Approve Software Request
Provide User With Instructions To Download Software
Download and Configure Software To User Device
Bill User For Software Requested
To:
Fill Out and Submit Software Request Form
Validate Software Request Form
Provide User With Instructions To Download Software
Download and Configure Software To User Device
Bill User For Software Requested
Approve Software Request
Grant user software right away and get approvals in the background
If user not authorized, then remove the user's access to the software

In this example, getting management approval for software access can introduce days or even weeks of delays. Here, access to the software is automatically granted. If it is determined at a later time that the user shouldn't get the software, then their access is taken away.

Strategy #4 – Make Step Activities Move Faster (Increase Their Velocity)

With this strategy, you are not really taking anything out, but you are introducing automation, batching or other mechanisms to operate the step much faster.


From:
20 min
30 min
2 Hours
Fill Out and Submit Software Request Form
Validate Software Request Form
Provide User With Instructions To Download Software
Download and Configure Software To User Device
Bill User For Software Requested
Approve Software Request
To:
1 min
1 min
10 min
Fill Out and Submit Software Request Form
Validate Software Request Form
Provide User With Instructions To Download Software
Download and Configure Software To User Device
Bill User For Software Requested
Approve Software Request
User selects from desktop catalog of software products
User is only notified that software installation will start
Download is "pushed" and configured once user selects software

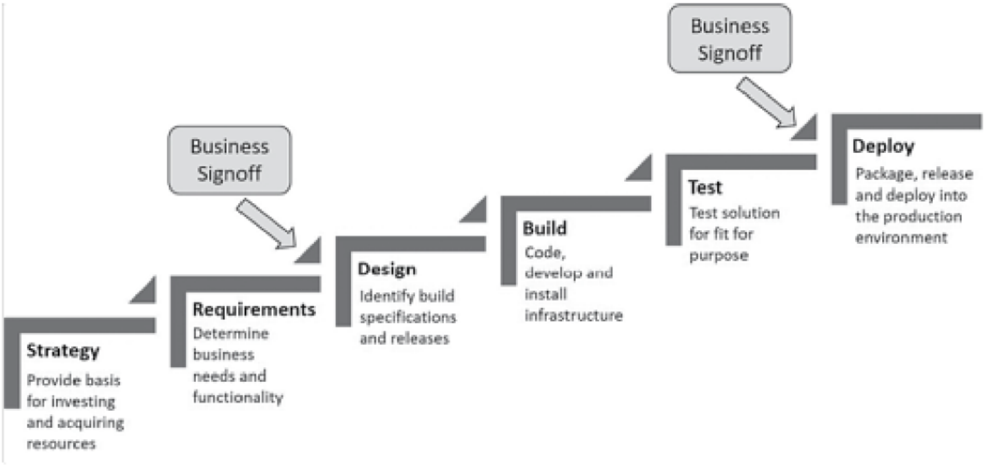
With the above, a strategy has been employed to utilize an automated software catalog. The user selects the desired software from the catalog. Once selected, the software will download and configure the software to the user’s device.

Chapter 4

Agile Principles Applied To ITSM

Overview Of Agile

IT, for decades, has employed a development lifecycle for building IT solutions known as the Waterfall Development Model. Behold this in all its glory:



Business Signoff
Business Signoff
Strategy
Provide basis for investing and acquiring resources
Requirements
Determine business needs and functionality
Design
Identify build specifications and releases
Build
Code, develop and install infrastructure
Test
Test solution for fit for purpose
Deploy
Package, release and deploy into the production environment

While this model has served IT well, it presents several problems in today's fast moving business climate. It's problematic for the following reasons:

It promotes distrust and bad relationships with the business. The Business is skittish to sign off because they won't see anything else until the next release cycle. To cope with this, the business slows down IT by making sure every requirement they absolutely need is included. This is because they know there is a long period of time until they get to influence the solution again.

It's slow and highly wasteful.
There is a long gap between when the business signs off on the solution and when it is ready to go live. In fast moving business climate, there is a good chance that business conditions may have changed such that the original requirements may no longer be valid or are now incomplete. The business again slows down IT from releasing the solution because requirements have changed. Now IT has rework to do because some requirements have changed.

It's risk prone.
The business essentially walks away between the time they have signed off on requirements and when they come back for user acceptance and sign off. IT is essentially throwing the dice that they implemented everything the business needed. That they didn't misinterpret what they thought the business wanted. That the solution will be understood and usable by business stakeholders. That the solution can be operated day-to-day at acceptable cost and risk. When expectations are missed, it leads to either significant delays and costs or scrapping the effort altogether losing their entire investment in the solution.

It introduces a deadly cycle of complex big-bang releases and technology roadmaps.
Picture this scenario: Business Unit gets a release of a new application in which that release is dependent on release X of a database which in turn is dependent on release Y of an operating system. Throw in a dependency on a release of new desktop clients. Now everyone is locked in. The desktop may not be able to be upgraded because of negative impacts to the application. The application may not be able to be upgraded because of dependencies on the OS or database. To cope with all; this, technology roadmaps, upgrade schedules and dependent projects need to be developed. Everything just takes longer and longer.

Agile is an evolving new practice for IT in which development activities start before all the solution requirements are known. Rather than develop an entire solution before release, solutions are constructed in iterations of small increments that are bundled and released to the business on a much shorter frequent basis. As the business gets one release, the development team is already enhancing that solution with further features and improvements. The business gets to see, touch and provide feedback on solutions frequently for every release. They no longer have to wait months to see what they are getting.

Here is a living example from an actual large bank. The bank's lending division was overhauling their business intelligence reporting with all new reports, dashboards and analytics. The bank chose to use an agile approach to develop their solution. Development activities looked as follows:

1. Development teams met with bank lending representatives to identify data requirements and mocked up a small set of reports
2. The development team looked at the tasks they needed to do and created a backlog of work upon which they would prioritize and execute
3. This backlog was quickly chunked into "sprints" which represented short small releases that could be accomplished by the team
4. A first "sprint" release of the data model and subset of the reports was produced in 2 weeks
5. The representatives were called in to view the reports and provide feedback
6. Feedback was provided again by representatives directly to the development team
7. This added to the backlog and enhancements were added for the next "sprint" which was released 2 weeks later
8. Steps 5, 6 and 7 were then repeated about 3-4 more times until the backlog was greatly reduced and eventually completed

So what was the end result? Business representatives saw a working solution in just 2 weeks. Expectations were set such that they knew not everything would be included in the first release. They got to provide input all along the way. Satisfaction was high. Delivery was quick.

What would have happened if a Waterfall approach had been taken? Most likely weeks and weeks of user interviews and requirements gathering. Anxiety over signing off on a horde of requirements. Months of development activity for all those requirements. Missed expectations for reports that looked different than what they had expected them to be. Months go by. No value is delivered. How do we know that could happen? It actually did – with another development vendor prior to redoing the effort with the agile approach previously described.

Agile Concepts

So what does it mean to be agile when developing solutions?

Agile development is a means and set of solution development principles in which requirements and solutions evolve through team collaboration, adaptive planning, early customer feedback and continuous improvement. The key difference with Agile from past development practices is that the end state solution is not fully known at the time you start building and deploying. Rather than hold back build efforts until all requirements and design are complete, you start with an early set of requirements, build on those, get customer feedback, then update your requirements, build some more and so on until the full end state solution is reached.

Agile practices are not new. Their roots go back to the mid-1980s although incremental development methods themselves go back as far as the late 1950s. Over concerns about software development being too cumbersome and slow, a Manifesto for Agile Software Development was developed by the DSDM (Dynamic Systems Development Method) during the late 1990s and formally proclaimed in 2001. The Manifesto railed against highly regulated and regimented Waterfall methods and advocated a return to more lightweight and rapid development approaches. It's goal; was to promote better ways of developing software and valued customers, people, collaboration and rapid change over traditional documentation, contract negotiation, and following a strict plan.

A number of agile methods existed prior to publication of the Manifesto. These included practices such as DSDM XP, and

Scrum developed during the 1990s.

Some of the key concepts of agile present interesting ways of working that are different than IT may be used to working with:

- Requirements are not fully known before you start building – they evolve over a continuous loop of customer interaction with the solution and feedback
- Customer interaction with the software as it is being constructed provides much more accurate requirements and greatly increases customer confidence in the solution than using interviews, documenting requirements in a presentation and asking for customer signoff
- Demonstrating working software to customers goes much farther than just presenting documents in a meeting
- Requirements are elaborated as Features and User Stories and closely tracked by Teams typically using Kanban Boards or other similar approaches and displayed publicly
- Daily “stand-up” meetings (also known as the daily scrum) are held to report progress toward the sprint goal, what will be done that day, and any roadblocks that might exist.
- Continuous development is promoted with a sharp focus on rapid response to change

Agile development methods break solutions down into small increments (Sometimes called Features and User Stories). These increments are then assigned to Iterations (short time frames called Sprints) that typically last from one to four weeks. Each iteration involves a cross-functional team working on all functions from planning to test. At the end of the iteration or Sprint, a working product is demonstrated to stakeholders. This greatly reduces risk and allows the project to adapt to changes quickly. An iteration does not produce a fully complete solution, but has enough functionality to be released to the customer. Multiple iterations might be required to fully release a product or new features. The Agile Operating Model presented later in this chapter shows a general approach that can be adopted by IT organizations.

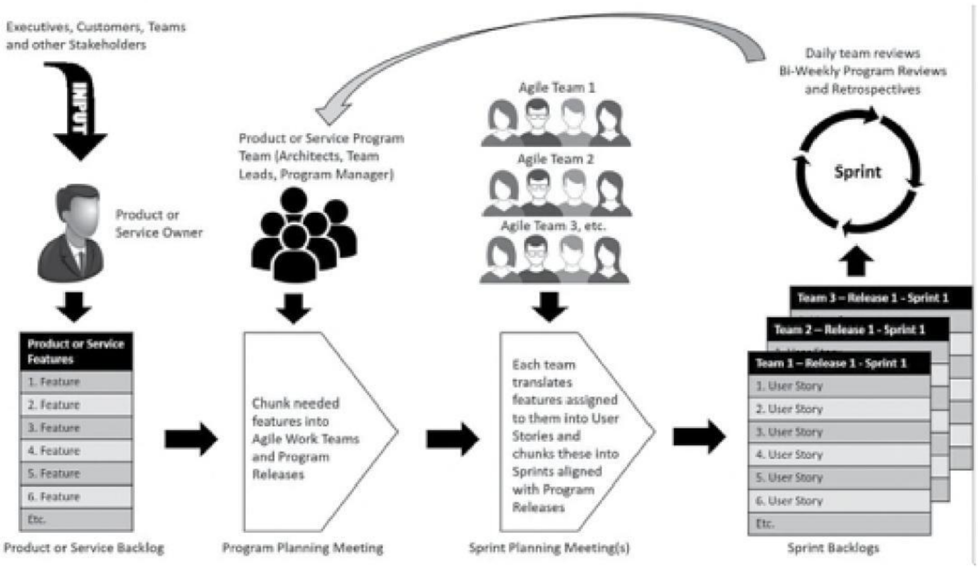
An Agile Operating Model For ITSM

This section presents an overall operating model that can be used to build and develop services and products in the context of IT Service Management. In IT organizations, development of solutions doesn't start and end with applications. It includes those applications in an eco-sphere that is surrounded by supporting infrastructure, operations, Service Desk and other support functions. Release of an application by itself is problematic if these latter functions are not configured to operate the application at acceptable cost and risks to the business.

As a result, many projects will include not just the application development team, but also operational teams. Operating High Velocity ITSM includes the use of DevOps. Therefore the implication is that solutions under development will always have at a minimum, two teams: Development and Operations.

The model presents a way for how multiple teams may extend an agile development approach to accommodate use of multiple teams. Essentially this is having each team individually follow agile practices and working in Sprints. The team Sprints themselves are aligned and coordinated across all teams into a planned Release. This sometimes has been referred to as “Scrum of Scrums” in terms of its approach.

The model is presented as follows:


Executives, Customers, Teams and other Stakeholders
INPUT
Product or Service Owner
Product or Service Features
1. Feature
2. Feature
3. Feature
4. Feature
5. Feature
6. Feature
Etc.
Product or Service Backlog
Product or Service Program Team (Architects, Team Leads, Program Manager)
Chunk needed features into Agile Work Teams and Program Releases
Program Planning Meeting
Agile Team 1
Agile Team 2
Agile Team 3, etc.
Each team translates features assigned to them into User Stories and chunks these into Sprints aligned with Program Releases
Sprint Planning Meeting(s)
Daily team reviews
Bi-Weekly Program Reviews and Retrospectives
Sprint
Team 3 – Release 1 - Sprint 1
Team 2 – Release 1 - Sprint 1
Team 1 – Release 1 - Sprint 1
1. User Story
2. User Story
3. User Story
4. User Story
5. User Story
6. User Story
Etc.
Sprint Backlogs

Let's follow the model above in counter-clockwise fashion starting at the upper left:

A service (or product) starts with Service Strategy. Executives, meet with customers and other stakeholders to determine what is needed. From this, they determine a service or product position and what key business outcomes they will be investing in.

A Service Owner (or Product Owner) is then assigned. This role reviews the strategy and creates a vision of the service. Key features are identified and inventoried. This inventory becomes the Service (or Product) Backlog.

At this point, a Service Team is pulled together. This team consists of an overall Program Manager, Architects and Team Leads. Note that many services are usually dependent on other supporting services to operate. As an example, an Email Service depends on supporting services like Network Management, Server Management, etc. Therefore, the Program Team will bring in a Team Lead person from each dependent service as well as a Team Lead for the Service being constructed. Once pulled together, a Program Planning Meeting is established to:

Continuous Release (described later in this book). Features are validated for the next release and off everyone goes again.

More formal methods are available in the industry that address use of Agile for larger projects and multiple teams. Examples of these include:

- Scaled Agile Framework (SAFe)
- Disciplined Agile Delivery (DAD)
- Large-scale scrum (LeSS)
- Nexus (scaled professional Scrum)
- Scrum at Scale
- Enterprise Scrum
- Setchu (Scrum-based lightweight framework)
- Xscale

Infrastructure Projects Can Also Be Agile

There is a misconception that agile development approaches can only be used for application development projects. That it doesn't apply for infrastructure projects. In actuality, almost any development or infrastructure effort can be done as an agile project. It still comes down to the Features and User Stories that get built and deciding which Sprint those features and User Stories will reside in.

Let's look at an example of how this might be done. The desire here is to implement a new Asset Management solution. Needed here is a new tool, new processes, operating roles and responsibilities and an effort to migrate current asset data to the new system.

Here is how the Sprints for such an effort might be laid out:

Sprint	Scope
01	Install out of box hardware and setup network connections including connection to the cloud provider

	capabilities to bulk load assets
11	Build asset export features and test these to see if assets can be downloaded to other venues such as a text file or spreadsheet
12	Build out any custom queries for searching and finding assets – again execute with use cases to help drive what kinds of queries should be built and made public to users
13	Install support infrastructure such as service desk support categories and linkage to service desk tools to support new solution
14	Reserve this Sprint for training users and support staff for the new asset management solution
15	Reserve this Sprint for deployment and operational readiness activities – then deploy the solution out to production

Notice the agility being employed here versus a typical infrastructure Waterfall Project. Assuming each Sprint is about 1 week long, business users are brought in as early as Week 3 to start seeing a working solution instead of waiting months and months before they see anything. The solution is constructed in small increments which focuses implementation teams. Management can see progress by looking at progress within each Sprint.

The point here is that anything can go agile. You just simply identify your solution backlog and then chunk it out into sprints. No need to treat infrastructure projects any differently than application development projects.

Maintenance and Support Can Be Agile

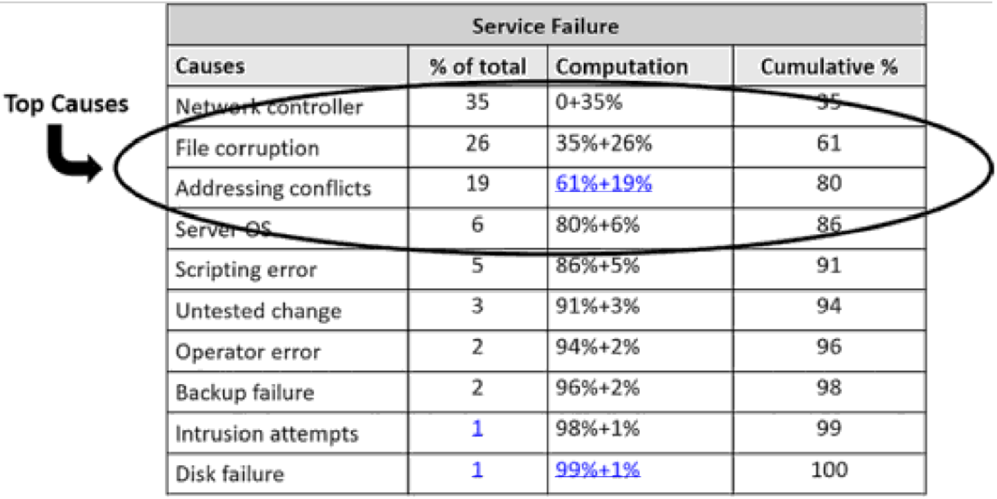
IT Service Management is not just about building and managing IT services, it also includes the ongoing daily support for handling incidents, problems, application management, and patching of hardware and software. Since these are not really building and developing, can they be handled with agile approaches?

The answer is yes. Here is how one large government agency took on this task:

1) Incidents, problems, bugs and patches were classified into the services that were impacted
2) Support teams were assigned to “own” each incident, problem, bug or patch
3) Tasks to fix software and hardware issues were placed and managed on a separate Kanban Board for each team
4) Each team organized their assigned tasks into Sprints – meaning they were grouped into what Sprint they would be fixed in
5) Sprints were only 1 week in duration
6) Larger tasks were broken down into sub-tasks if the teams determined that the main task could not be accomplished in a single Sprint
7) Daily standup meetings were held with each team to track progress

Prior to using the above approach, the support teams were inundated with hundreds of bugs, patches and requests for a new service that had just been deployed. Support and technical teams were overwhelmed with the backlog and struggled to prioritize and find a way to effectively deal with all the work. Users of the service were extremely frustrated and questioned whether it should have rolled out in the first place.

Once the Agile approach was introduced, things moved fast. Batching tasks into Sprints allowed for more work to be completed faster. Users saw the progress on a Sprint by Sprint basis and their



Service Failure			
Causes	% of total	Computation	Cumulative %
Network controller	35	0+35%	35
File corruption	26	35%+26%	61
Addressing conflicts	19	61%+19%	80
Server OS	6	80%+6%	86
Scripting error	5	86%+5%	91
Untested change	3	91%+3%	94
Operator error	2	94%+2%	96
Backup failure	2	96%+2%	98
Intrusion attempts	1	98%+1%	99
Disk failure	1	99%+1%	100