



Using ITIL to address the disconnect between development and IT service management teams.



June 2008

Contents	
2	Introduction
3	Part 1: ITIL and software development
4	From silos to conceptual disconnects
6	Why collaborate?
7	Establishing governance
8	Part 2: Making processes actionable
11	Mapping processes
11	Identifying key interface areas
14	Summary
15	For more information
15	About IBM Service Management

Introduction

The failure of software to sometimes fulfill its intended goal is as old as software itself. Unlike other engineering disciplines, software engineering does not have to deal with inherent physical and/or temporal constraints.* Software is infinitely malleable – which can be both a blessing and a curse. Without constraints, it is easy for software to not fulfill expectations by failing to incorporate key requirements, including security, compliance and end-user specifications.

Now consider these implications in light of modern business practices in which virtually every business process, somewhere along its course, depends on software. In fact, technology advances have helped organizations achieve differentiation and customer value that could not have been possible otherwise. But with this dependence on software comes even greater responsibility. It is crucial for organizations to optimize software and service delivery processes to ensure the availability of timely, cost-effective, reliable and – most importantly – business-aligned software.

Historically, IT has included two domains: development and IT operations. Development created the software that was installed and operations kept it running. With little interaction between the two aside from the handoff, different cultures and communities have grown up in the development and operations domains.

Frameworks such as IT Infrastructure Library® (ITIL®), however, have brought renewed vigor to IT operations, effectively transforming it into IT service management. As part of this increased focus on providing cost-effective IT services that meet customer needs, IT service management must work more closely with development to help ensure the IT services meet service level targets and other customer expectations.

Highlights

This paper uses ITIL as a framework to identify where and how development and IT operations teams are codependent and how this relationship can be improved. The paper then compares actionable representations of ITIL with specific software development processes and identifies specific actions that better align development and IT operations, to help improve the effectiveness of software and service delivery.

Part 1: ITIL and software development

First released in 1988, ITIL is a library of books that document vendor-neutral industry best practices to help organizations better manage their IT services – or rather, IT service management. The books identify, in great detail, the activities considered fundamental to the delivery of high-quality, business-aligned IT services – where quality runs the gamut of risks an IT organization must face, from security concerns to compliance, business continuity and end-user satisfaction. Far more than just an application, an IT service encompasses systems, people and consumable resources. It is the lifeblood of an organization, the means by which organizations meet the primary expectations of their customers, suppliers and business partners.

An IT service encompasses systems, people and consumable resources; it is the lifeblood of an organization, the means by which organizations meet the primary expectations of their customers, suppliers and business partners

Now in its third release, ITIL comprises five books, each detailing one of the five phases of the service life cycle. They are:

- **Service Strategy** — determining the overall strategy for how IT services will be provided.
- **Service Design** — designing IT services to meet service level targets.
- **Service Transition** — moving IT services from development to operational status.
- **Service Operation** — executing and supporting IT services.
- **Continual Service Improvement** — enhancing IT services based on past performance.

Highlights

The service life cycle is the primary theme of ITIL, Version 3. This life cycle captures not just the life cycle of a service but also indirectly captures the connection points linking development and IT operations processes.

Having an effective IT service life cycle requires efficiency in all phases, including Service Design, which discusses the design and development of IT services and, thus, includes development concerns. This recognition is one of the notable changes introduced by ITIL, Version 3. While Version 2 addressed software development as a more tangential concern in a secondary book, Version 3 identifies software development in a core book as an indispensable aspect of delivering effective IT services.

From silos to conceptual disconnects

Over the years, software development processes and IT operations processes have operated as completely separate entities within the organization. Software development processes can range from classic waterfall and V-model frameworks to well-defined iterative frameworks like IBM Rational Unified Process® (RUP®). IT operations processes are best understood as those identified within ITIL, International Organization for Standardization (ISO) 9000 or enhanced Telecom Operations Map (eTOM).

The independence of these two domains has effectively created silos, with various process models, terminology, accountability measures and products specific to the domains. The two communities are often unaware that such disconnects exist, and this makes for considerable confusion when teaming is needed.

The development and IT service management domains share a number of concepts and terminology; yet, both domains often have a different understanding of what those terms mean

For example, the development and IT service management domains share a number of concepts and terminology. Yet, both domains often have a different understanding of what those terms mean. The most prominent of these are indicated on page 5.

Concept/term	Meaning in development	Meaning in IT service management
Configuration item	An item that makes up the system under development	An item in the IT infrastructure
Configuration management system (CMS)	A system that controls various versions of software (and hardware) items that make up the system under development, allowing check-in and check-out as needed	A large virtual system that identifies all controlled items in the IT infrastructure as well as various media libraries and hardware stores
Configuration management	Controlling the artifacts within a development project	Controlling configuration items within the IT infrastructure
Change management	Overseeing changes to a system under development	Overseeing changes to systems in the operational IT infrastructure
Release	A stable, executable version of a developed product	Collecting configuration items that are used to implement a change to an IT service
Release management	Making a developed system available to be deployed within an IT infrastructure	Deploying a released set of configuration items to targeted parts of the IT infrastructure

Terminology disconnects can result in related or overlapping processes that conflict rather than complement each other as they take different approaches despite seemingly shared objectives. In real terms, this means the development and IT service management teams can be poorly coordinated, putting software and service delivery at risk. With business success depending more and more on IT execution, organizations must find a way to bridge these silos and to promote more effective collaboration between the development and IT service management teams.

Highlights

Functional requirements describe the things a service is intended to do and can be expressed as tasks or functions that a service is required to perform; nonfunctional requirements concern constraints on the system — usability, reliability, availability, capacity and more

Why collaborate?

A good way to think about the value of collaboration between development and IT service management is to consider the service's requirements. It has already been established that successful businesses depend on high-quality IT services. But what defines "quality"? The answer is the fulfillment of a service's requirements. And as you will see, the fulfillment of certain requirements depends on collaboration.

Requirements are typically divided into two categories: functional and non-functional. Functional requirements are more familiar. They describe the things a service is intended to do and can be expressed as tasks or functions that a service is required to perform. Such requirements can be expressed at either a very high level — for example, "handle orders for office supplies" — or can be expressed at a very low, detailed level — for example, "allow users to order office supplies from a catalog, place them in a virtual shopping cart and check out." In the case of functional requirements, "quality" is a yes-or-no question — does the service perform its function or doesn't it?

This narrow definition of quality, however, omits the other kind of requirement: nonfunctional requirements. Nonfunctional requirements concern constraints on the system — usability, reliability, availability, capacity and more. This is not about *what* tasks the service is performing but about *how well* it is performing. In this instance, the judgment of quality leads to a more satisfying definition: fitness for use. Understandably, a system that fulfills its functional requirements is much more desirable than buggy software that fails utterly. However, if fulfillment of those functional requirements takes minutes to hours while end-user expectations are response times measured in seconds, the service still remains a failure.

Unlike functional requirements that are almost solely the responsibility of the development team, both software development and IT operational teams own responsibility for fulfillment of these nonfunctional requirements. As an example, service level agreements (SLAs) specify minimal levels of availability, capacity and other nonfunctional characteristics indicative of a well-executed service. Because both development and IT operational teams share responsibility for fulfilling the SLA, these teams must collaborate. This impacts both the development and release of new capabilities as well as the identification and repair of errors found in the live production environment. There is an implied movement and sharing of data that can only occur if development and IT service management teams are collaborating effectively.

Establishing governance

In order to promote effective collaboration, the IT organization must institute governance policies to define the interaction between the processes in these domains. The two teams traditionally have not collaborated, so these governance considerations are often overlooked. If there's no collaboration, there are no collaboration requirements. Yet collaboration always occurs, whether IT organizations explicitly embrace the notion or not. The question is simply how well the collaboration occurs.

For example, consider the release process. In simple terms, development builds the software that underlies a particular service, and IT service management deploys that software into the production environment. But there are other aspects that must be defined:

- How exactly does development hand-off software to IT service management?
- How does IT operations verify that the right software has been delivered?
- Is the targeted IT infrastructure properly provisioned to support the software?
- Has the software passed all required verification tests?

Highlights

Good collaboration requires governance — with effective oversight and control that holds each team and member accountable for their unique role in the process

The reality is that this process is far from simple. Instead, it is a complex, error-prone process that can easily undermine the quality of the business process(es) depending on the released software. To believe that good collaboration between development and IT service management is not important dooms such a process to failure. And good collaboration requires governance — with effective oversight and control that holds each team and member accountable for their unique role in the process.

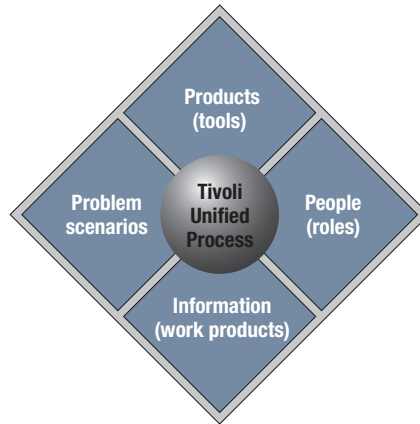
Accepted across the world as a set of best practices for IT, ITIL is a useful tool for understanding the elements of good collaboration. The following sections of this paper illustrate the policies of collaboration that could be instituted by appropriate governance, and discuss how to make ITIL actionable for processes in general, those particularly suited to collaboration between IT service management and development teams, and key interface areas between development and service management.

Part 2: Making processes actionable

ITIL is invaluable as a set of best practices, yet these practices must be made actionable in order to understand the connection and therefore collaboration opportunities between development and service management. To be actionable, the practices should be organized, ordered and sequenced into detailed workflows. Once these workflows are available, an enterprise can carry out IT service management best practices as a harmonized whole. The workflows should be correlated with the following:

- **Roles** — the parts individuals play in each task
- **Work products** — the inputs and outputs of each task
- **Tools guidance** — how to use tools to perform the tasks
- **Scenarios** — how real-world IT work uses parts of various processes

One of the best sources of this information is IBM Tivoli® Unified Process – a Web-based tool that makes ITIL actionable by including process workflows and the elements described previously. Using a Web browser, you can quickly click through various levels of processes, activities and tasks that show how to carry out the best practices. In addition, Tivoli Unified Process shows how these workflows map to well-known industry reference models such as Capability Maturity Model Integration (CMMI), Control Objectives for Information and related Technology (COBIT), Sarbanes-Oxley and ISO standards.

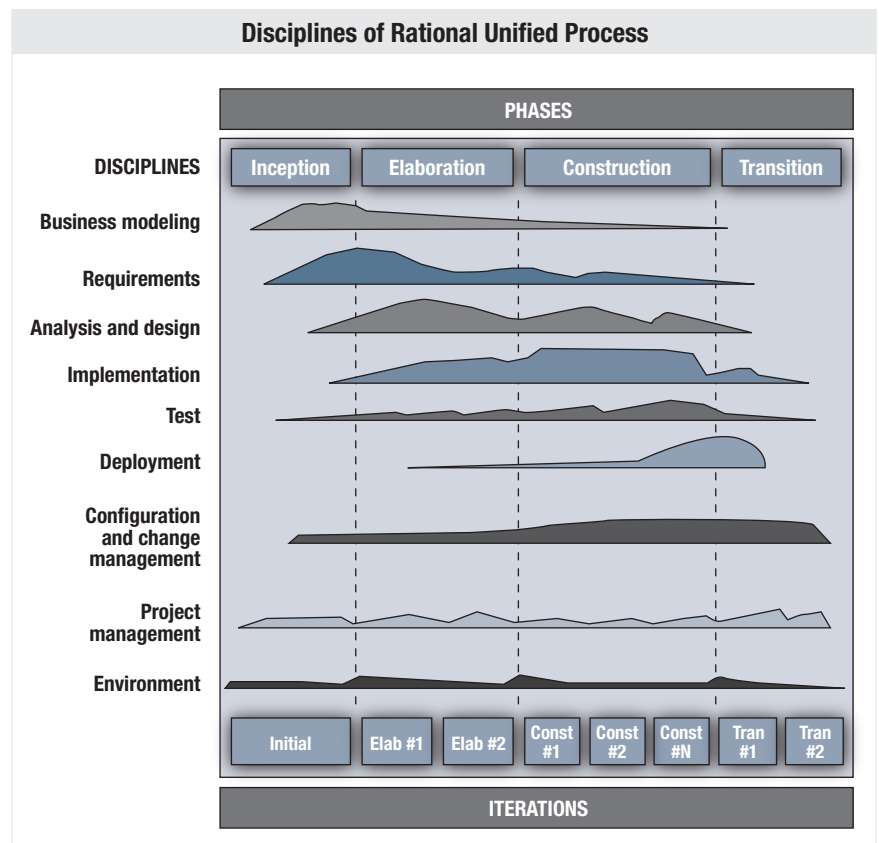


The contents of Tivoli Unified Process

Tivoli Unified Process contains 46 processes divided into eight process categories. Tivoli Unified Process is broad in scope and includes all of IT, including IT service management, IT development and IT administration. However, it only covers IT service management in depth and discusses development in generic terms in order for existing mature software development life cycles (SDLCs), such as RUP, to be used instead.

RUP is the counterpart to Tivoli Unified Process for IT development. This set of integrated workflows and related content has a great deal of maturity, legacy and user community support because of the length of time in which it has existed. (RUP can be thought of as the grandparent of Tivoli Unified Process and related endeavors.)

There are a variety of RUP customizations, including classic RUP, IBM RUP for Large Projects, IBM RUP for Maintenance Projects and others. The primary construct of RUP is a discipline, not a process. The following chart shows the disciplines of RUP.



Highlights

Mapping processes

Tivoli Unified Process contains processes that overlap with content areas (processes) in RUP. This allows the development area of Tivoli Unified Process to easily map to RUP, while allowing other SDLCs to map to Tivoli Unified Process as well.

The following table shows the areas of Tivoli Unified Process that map to areas of RUP. Knowledge of this mapping allows you to virtually “swap” an RUP discipline into Tivoli Unified Process, enabling you to bring the mature RUP content to bear within Tivoli Unified Process.

Tivoli Unified Process	RUP discipline
Solution requirements	Requirements
Solution analysis and design	Analysis and design
Solution development and integration	Implementation
Solution test	Test
Program and project management	Project management
Deployment management	Deployment

Identifying key interface areas

Apart from areas of Tivoli Unified Process and RUP that overlap, there are areas of Tivoli Unified Process and RUP that do not overlap but have important interfaces. There are at least four key interface areas between the development and service management domains. The graphic on the next page depicts these key interface areas.

There are at least four key interface areas between the development and service management domains

INTERFACE AREA	TIVOLI UNIFIED PROCESS	RATIONAL UNIFIED PROCESS
Design for service level targets	<ul style="list-style-type: none"> • Service level management • Availability management • Capacity management • Security management • IT service continuity management • Facilities management • Compliance management 	<ul style="list-style-type: none"> • Analysis and design
Managing changes	<ul style="list-style-type: none"> • Change management 	<ul style="list-style-type: none"> • Change and configuration management
Deploying a software release	<ul style="list-style-type: none"> • Release management • Deployment management 	<ul style="list-style-type: none"> • Deployment
Support for deployed software	<ul style="list-style-type: none"> • Incident management • Problem management 	<ul style="list-style-type: none"> • Deployment

Key interface areas between Tivoli Unified Process and RUP

The first area is *design for service level targets*. This area corresponds to the ITIL Service Design phase. Service delivery managers in IT service management must provide IT services that meet customer service level targets, which typically include targets for availability (such as uptime) and capacity (such as the number of users that can be supported). Software applications must be developed with expected service level targets in mind. Limitations on expected service level targets must be made known to IT service management. In addition, software applications that are instrumented for monitoring are much easier to manage concerning their service level targets.

The second area of integration is the *management of changes*. This can be found in the ITIL Service Transition phase. IT service management and development teams typically manage their changes separately. However, there often is a need to integrate change management between the two domains. For instance, if a user requests new functionality of an IT service, this request may first be considered by IT service management before it is considered by the appropriate development team. Approval and implementation of such a change must be coordinated between service management and development.

Next, there should be communication between development and service management concerning the preferred approach to *deploying a software release*. This is also found in the Service Transition phase. Will end users install the release? Will the release be deployed via automated push deployment? Are there complex prerequisites that will need to be installed? These types of considerations must be communicated between the two domains.

Finally, *support for deployed software* must be considered. This is addressed by the ITIL Service Operation phase. The first level of application support is typically not resident within a development organization but within the IT service desk. The highest level of application support, however, is typically provided by the development organization. Careful coordination must take place between the service desk and development to avoid letting incidents fall through the cracks. In addition, errors discovered during development must be made known to the IT service desk. Similarly, incidents and problems discovered by the service desk must be made known to the development team to fix in subsequent releases.

As an example of the interface between development and service management, consider the scenario in which an application development project

Highlights

Business success depends on IT execution — and IT execution depends on the successful collaboration of the development and IT operational teams

tester has identified known errors that were not fixed before development was completed. This information must be made available to support personnel who may have to deal with known application errors.

In RUP, this information would have been identified during testing. During the RUP Deployment discipline, error information would be recorded in User Support Material by the Deployment Manager. This work product would then be provided to the IT service desk. The IT organization would consider this to be an Operational Documentation work product and would refer to that information by the Incident Analyst or the Problem Analyst (during Incident Management or Problem Management, respectively).

Summary

To an ever-greater extent, business success depends on IT execution — and IT execution depends on the successful collaboration of the development and IT operational teams. Yet these two teams have historically operated independently, with process models, terminology, accountability measures and products specific to each. This cultural isolation threatens an organization's ability to successfully fulfill the myriad requirements for high-quality IT services. At the same time, a lack of oversight and control can make it difficult to properly govern and establish accountability for critical processes that require collaboration from both IT and development teams.

ITIL, the leading service management best-practices framework, offers a useful tool for understanding how to improve collaboration between the two. Tivoli Unified Process and RUP deliver a practical way to make ITIL actionable through detailed workflows correlated with roles, work products, tools guidance and scenarios. With the help of frameworks like ITIL and processes like Tivoli Unified Process and RUP, organizations can tackle the disconnect

between development and IT service management. The end goal of increased collaboration and harmony within the IT organization can bring with it the ability to deliver high-quality, business-aligned services that more than meet customer expectations.

For more information

To learn more about Tivoli Unified Process and RUP, contact your IBM representative or IBM Business Partner, or visit ibm.com/itsolutions/servicemanagement

About IBM Service Management

IBM Service Management helps organizations deliver quality service that is effectively managed, continuous and secure for users, customers and partners. Organizations of every size can leverage IBM services, software and hardware to plan, execute and manage initiatives for service and asset management, security and business resilience. Flexible, modular offerings span business management, IT development and IT operations and draw on extensive customer experience, best practices and open standards-based technology. IBM acts as a strategic partner to help customers implement the right solutions to achieve rapid business results and accelerate business growth.



© Copyright IBM Corporation 2008

IBM Corporation
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
June 2008
All Rights Reserved

IBM, the IBM logo, ibm.com, Rational Unified Process, RUP, Tivoli and Visibility. Control. Automation. are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

Other company, product and service names may be trademarks or service marks of others.

Disclaimer: The customer is responsible for ensuring compliance with legal requirements. It is the customer's sole responsibility to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the reader may have to take to comply with such laws. IBM does not provide legal advice or warrant that its services or products will ensure that the customer is in compliance with any law or regulation.

*The intended target for the software may have deep physical or temporal constraints, like medical devices or automotive instruments. Even in these cases, however, the software does not possess its own limitations. A developer can craft anything he or she chooses. Whether or not the software is suitable for the intended device is a separate issue.