ITIL
pragmatic and simple

Those who have been wrestling with the ITIL books may appreciate the simplicity presented in this article.

Geir Isene

g@isene.com

http://isene.com

and

Brendan Martin

brendan.paul.martin@gmail.com

inspired by ITIL projects at

The Norwegian Olympic and Paralympic Committee and Confederation of Sports

The Norwegian Meteorological Institute

Norwegian Business School

Telenor ASA

Ventelo AS

Telemark County

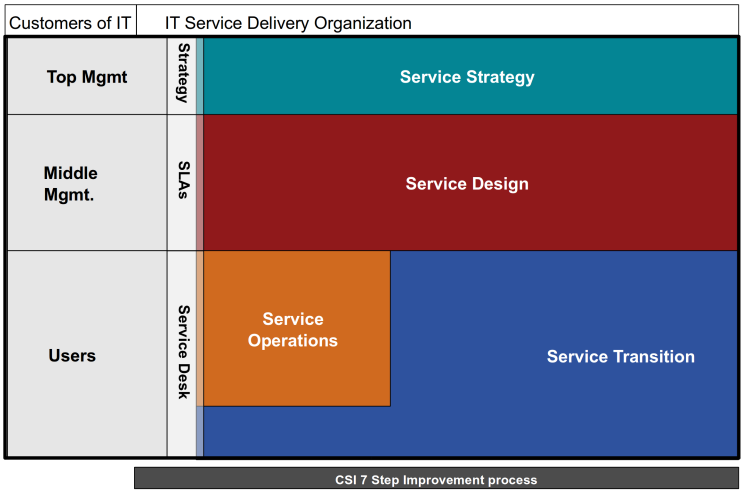
Bohus AS

Color Line AS

editing

Marilyn Abrahamian

Version: 2026-11-10



Document link: http://isene.com/

Contents

1 Introduction 3

2 Service Desk 5

3 Request Fulfillment 7

4 Incident Management 8

5 Problem Management 9

6 Change Management 10

7 Release & Deployment Management and Validation & Testing Management 11

8 Service Assets & Configuration Management 12

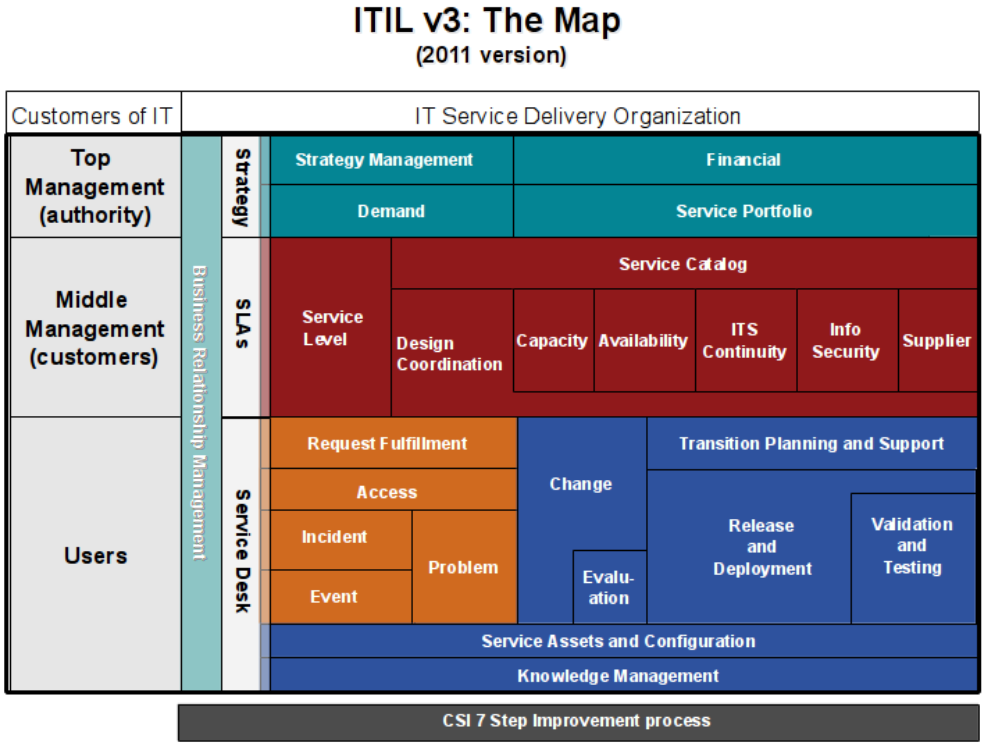
9 Knowledge Management 13

10 How to proceed 14

Abstract

The IT Infrastructure Library[1] (ITIL) is a best practice framework for service management[2]. ITIL is a trademark owned by the Cabinet Office (part of HM Government). The framework was originally intended to serve the delivery of Information Technology within a company, but can also be used outside of IT and for delivery of services between companies. It could be argued that ITIL focuses on the implementation of processes to facilitate service management.

This article provides a pragmatic view of ITIL – a simpler and more straightforward implementation of some of the core ITIL processes. The ITIL maps provided in this document are the copyright of FreeCode AS and are licensed under the GPL v3[3] license.



[1] ITIL: http://www.itil-officialsite.com/
[2] Service management: http://en.wikipedia.org/wiki/Service_Management
[3] GPL version 3: http://www.gnu.org/licenses/gpl-3.0.txt

1 Introduction

For an introduction to the ITIL[4] framework, WikiPedia provides an excellent starting point.

The purpose of this article is to provide the shortest possible description of an alternative implementation of some of the most central ITIL processes. It should be noted that the way ITIL sub-divides IT service management may not be the best way for a given company. There may indeed be other, better ways to organize and to share responsibility within an organization – especially when organizing other areas than the IT department.

Important basics of organization and responsibility are covered in the article titled, "Processes, Automation and Human Potential"[5], while a fresh look at organizing is presented in the article titled, "LiQUiD"[6].

ITIL splits responsibility into "responsibility" and "accountability"[7], where the "accountable" is responsible for the delivery and the "responsible" is responsible for following the process. However, since it is impossible to have a set process producing a set deliverable when the input is not fully known, this split often breaks down. The input to an ITIL process usually cannot be fully known. Therefore, the process must be flexible, to yield a determined output (deliverable). It is only with testing that you want a fixed process. All other processes should be adaptable, to accommodate for unmitigated input and yield the desired output.

The splitting of responsibility is unfortunate because it undermines a concept that is well understood in most societies and throughout history. We propose to dispense with this diluting of the concept and instead introduce the term "100% responsibility". This brings responsibility back to where it belongs. A person who is "100% responsible" for Change Management, for example, is fully responsible for *the deliverables* of Change Management. No excuses. No blaming of any "accountable" or the process. The person is charged with the responsibility of delivering the intended output. This obviously emphasizes the need to clearly define the authority of the person in charge of an area. It also stresses the need to assign responsibility to *the right person*.

Here, each ITIL process is presented with a short description, clearly defined deliverables with accompanying metrics, required process input and required authority for the person 100% responsible. A process is offered as a suggestion, but the person in charge is 100% responsible for the deliverables and must adapt the process to suit the organization, the culture and most importantly the desired output.

The central theme in any organization should be the concept of 100% responsibility. Add a focus on *simplicity* and *immediate relevance*, and you will have an organization with less overhead and more production of real, valuable results.

Remember that any policy, any rule, routine, process or procedure carries with it a certain cost – not only related to its implementation, but also the recurring cost every time it is applied, as well as the cost to maintain the structure it imposes on the organization. Sometimes this cost is outweighed by clear benefit, as delineated in the article, "Processes, Automation and Human Potential", but often it is not. Introducing bureaucracy for the sake of false security or pretended control and without first estimating its Net Present Value[8] constitutes a slippery slope of inefficiency.

It should also be noted that it is more important to be *effective* than it is to be *efficient*. To quote Peter Drucker:

> *"There is nothing so useless as doing efficiently that which should not be done at all."*

For this reason it behoves the organization well to focus first and foremost on delivering valuable results and secondly on efficient processes to best yield those deliverables.

The processes herein are presented using the HYPERLIST^S system. For a complete description of HYPERLIST^S , please read the full definition document[9].

Since HYPERLIST^S can contain both flows/transitions and descriptions/states, HYPERLIST^S can include anything and everything when describing ITIL processes. The processes to follow are fully explained – including not only the proposed process flows, but also a description of the process, defined inputs, deliverables and metrics.

A bare minimum description of HYPERLIST^S is given here, enough to understand the lists provided in this document:

- HYPERLIST^S contains Items (separate particulars, usually on a single line)
- An Item can have a "child", adding description or depth to its parent Item
- A "child" is an indented Item under its parent Item
- A semicolon (";") can be used instead of a line break to separate Items
- Top level Items are in bold for clarity

[4] ITIL introduction: http://en.wikipedia.org/wiki/ITIL
[5] Processes, Automation and Human Potential: http://isene.com/
[6] LiQUiD: http://isene.com/
[7] The RACI matrix: http://en.wikipedia.org/wiki/Responsibility_assignment_matrix
[8] Net Present Value (NPV): http://en.wikipedia.org/wiki/Net_present_value
[9] HYPERLIST^S : http://isene.com/

- An Item can contain various elements, color coded for clarity
- Square brackets indicate a condition or "Qualifier" for the Item (in green)
- Example of a Qualifier: [? Problem ticket needed] Create Problem ticket. In this example, you would only create a Problem ticket if the condition "Problem ticket needed" is met – i.e. if there is a need for a Problem ticket.
- A Qualifier starting with a question mark is read "If…" ([? Problem ticket needed] is read "If Problem ticked needed"
- A Qualifier not starting with a question mark is read "For…" ([Problem Manager] is read "For Problem Manager"
- A word in capital letters ending in a colon at the beginning of an Item is called an Operator (in blue)
- An Operator "operates" on that Item
- If a line ends in an Operator, it "operates" on all the Item's children
- Words not in capitals at the beginning of an Item and ending in a colon are a Tag (in red)
- Tags give structure information to an Item. If a line ends in a Tag, it applies to all the Item's children.
- Angle brackets ("<" and ">") indicates a Reference (in purple)
- A Reference in a process means that you jump to where the Reference points

Implementations of these processes as simple as they are represented here have proven very valuable in many companies and government organizations. While some may initially prefer to have the processes drawn as flow charts or other diagrams, it is easier to carry out collaborative work with pure text. And since HYPERLIST^S can accommodate both flows and states/descriptions, the whole process area can be described using such simple lists. There is perhaps nothing that cannot be described using HYPERLIST^S , and the full HYPERLIST^S document gives many examples of its usage, ranging from simple todo lists and shopping lists to project plans and breakdowns of logic structures.

2 Service Desk

Description
The IT Service Desk (SD) is the IT users' Single Point of Contact (SPOC)
SD receives all types of contact from IT users and registers issues
SD routes all issues to the appropriate recipients while informing the IT users of the progress
SD is responsible for all input channels from the IT users (portal, telephone system, etc.)
SD is responsible for the IT issue-tracking system and its usage
The term "Originator" is used for the IT user contacting the SD
Input
Contacts from IT users
Deliverables
Satisfied IT users
Service Request tickets
Incident tickets
Metrics
IT users satisfaction surveys
Correctly routed tickets (%)
Responsible
Service Desk Manager
Authority
[Service Desk Manager] To manage the Service Desk personnel
[Service Desk personnel] The use of common sense
Process
CONTINUOUS: Keep Originator informed of progress and update tickets as appropriate
Receive contact from an IT user (by phone, e-mail, self-service portal)
Create a ticket in the IT issue-tracking system describing the issue
Ensure the ticket is complete with the necessary information; Including:
Information on the IT user and the issue at hand from the user's perspective
Priority
Priority = Impact + Urgency (P = I + U)
Impact scale
1 = Non-critical affecting one user
2 = Non-critical affecting one department, or critical affecting one user
3 = Non-critical affecting one division, or critical affecting one department
4 = Non-critical affecting whole organization, or critical affecting one division
5 = Critical affecting the whole organization
Urgency scale
1 = Best effort
2 = Set specific time for resolution (more than one week)
3 = Resolution within one week
4 = Resolution within one business day
5 = Handle immediately
Category
Type of service
IT systems relevant for the issue
Type of issue
Incident
Service request
Request for information
[? Appropriate] Handle the issue immediately; <Close>
Assign the ticket to the appropriate recipient; OR:
Incident Management
Request Fulfillment
Appropriate process, department, role or person
Close
Ensure the ticket is appropriately closed

```
   Create needed documentation
Inform user and other appropriate roles
```

3 Request Fulfillment

Description
Request Fulfillment (RF) exists to handle Service Requests (SR) from IT users
RF executes pre-authorized (by Change Management) Standard Changes (StC)
RF helps the IT user submit a Request for Change (RfC) if appropriate
RF includes Access Management in this implementation
Input
Service Request tickets from SD
Deliverables
Service Requests handled according to the Service Level Agreement (SLA)
Metrics
Response time (according to SLA)
Resolution time (according to SLA)
Responsible
Request Fulfillment Manager
Authority
[Request Fulfillment Manager] To manage designated personnel so as to uphold SLAs
[Request Fulfillment personnel] The use of common sense
Process
Receive Service Request ticket from SD
[? SR is inappropriate] Reject the SR
<Service Desk/Process/Close>
Ensure the ticket is complete
[? Access request]
Ensure SR is handled in accordance with Information Security Policy
[? SR is not covered by an StC]
Help the IT user submit a complete RfC
Submit RfC
<Change Management>
Receive answer on RfC from Change Management
Inform IT user of the RfC decision
[? SR is covered by an StC]; OR:
[? RF cannot execute StC] Assign the ticket to the approved StC executor
[? RF can execute StC] Handle the SR according to the StC
Break down the delivery into logical parts
Assign each part of the delivery to the appropriate resources
Ensure each part of the delivery is completed
Assemble all parts into one complete delivery
Ensure the IT user receives the delivery
<Service Desk/Process/Close>

4 Incident Management

Description
Incident Management (IM) handles unplanned interruptions to IT services.
IM aims to help the IT user back into production as soon as possible
IM facilitates the restoration of production by the fastest possible means; OR:
Workarounds
Permanent solutions
Input
Incident tickets from SD
Deliverables
Incidents handled according to SLA
Metrics
Response time (according to SLA)
Resolution time (according to SLA)
Responsible
Incident Manager
Authority
[Incident Manager] To manage designated personnel so as to uphold SLAs
[Incident Management personnel] The use of common sense
Process
Receive Incident ticket from SD
Ensure the ticket is complete
[? Major Incident] <Process for Major Incident>
Assign ticket to appropriate resource
Analyze Incident; OR: (whichever is fastest)
Suggest and document a possible workaround
Suggest and document a possible permanent solution
[? No solution found]
Create Problem ticket
Keep the Incident open and go on with <Analyze Incident>
Implement WA or permanent solution
Verify with originator that the Incident is mitigated
[? Incident not mitigated] <Analyze Incident>
[? Problem ticket needed] Create Problem ticket
Ensure the appropriate documentation is entered in the Service Knowledge Management System (SKMS)
<Service Desk/Process/Close>

5 Problem Management

Description
Problem Management (PM) exists to find permanent solutions to underlying problems
PM aims to reduce the number of Incidents by eradicating the root cause of Incidents
PM acts reactively when handling tickets created (mostly by IM) and proactively by creating tickets
Input
[Reactive PM] Problem tickets (mostly created by IM)
[Proactive PM] Statistics/trends, creativity
Deliverables
Solutions to problems
Metrics
Recurring Incidents per IT user
Responsible
Problem Manager
Authority
[Problem Manager] To manage designated personnel so as to uphold SLAs
[Problem Management personnel] The use of common sense
Process
Problem ticket; OR:
Receive Problem ticket (mostly from IM)
Create Problem ticket (based on statistics/trends, experience and hunches)
Ensure the ticket is complete
Assign ticket to appropriate resource
[? No available and competent resource] Assign ticket to Service Responsible
Conduct a Root Cause Analysis
Suggest and document a possible permanent solution (or workaround)
[? Solution is not covered by an StC]
Submit RfC
<Change Management>
[? RfC not approved]; OR:
<Suggest and document a possible permanent solution>
Close the ticket as "won't fix"; <Problem Management/Process/Close>
[? RfC approved] Wait for the Release of the approved Change
[? Solution is covered by an StC]; OR:
[? PM cannot execute StC]
Assign the ticket to the approved StC executor
[? PM can execute StC]
Implement the solution according to the StC
Verify with the ticket creator that the Problem is solved
[? Problem is not solved] <Conduct a Root Cause Analysis>
Close
Ensure the ticket is appropriately closed

6 Change Management

Description
Change Management (CM) exists to develop, enhance and ensure stable IT services
CM aims to protect IT users from negative impact resulting from changes in IT
Input
Requests for Change (RfCs); From:
Request Fulfillment
Problem Management
Service Design
Other processes or departments
Deliverables
RfC decisions
Successful Changes
Metrics
Successful Changes (%)
Responsible
Change Manager
Authority
[Change Manager]
To manage designated personnel so as to uphold SLAs
The power to authorize or reject IT changes within the limits set by the Change policy
[Change Management personnel] The use of common sense
Process
Receive RfC
Ensure the RfC is complete; Including:
Submitter
Description of the required Change
Proposed Priority (Impact + Urgency)
Business Case for the Change; Including:
The reason for the Change (all the good arguments for the change are to be included)
Estimated cost (time/money)
Estimated benefits (increased revenue, profit, reduced cost)
The budget covering the cost
Cost limit (discard the change if the implementation will cost more than Y money)
Time limit (discard the change if the implementation will take more than X time)
Systems involved
Potential risk and negative consequences from doing or not doing the Change
Suggestions as to how the Change should best be done
Rollback plan, or a reason why this is not needed
Other information that will help the approver make a qualified decision
Evaluate the RfC according to the Change Policy; Including:
[? Impact <3 AND Competence >3] Individual resource can decide on the Change
[? Impact <3] Change Manager decides on the Change
[? Impact >=3] Change Manager decides on the Change after CAB (Change Advisory Board)
[? Change approved] Create approval for Work Package
Allocate time slot for deployment
Agree on the exact time within the time slot with Release & Deployment Management
Update Change Calendar
Ensure all relevant parties on the upcoming Change are informed
Send Work Package to <Release & Deployment and Validation & Testing>
PIR (Post Implementation Review)
Conduct a PIR according to the Change Policy
Ensure all relevant parties are informed
[? PIR failed]
Handle according to common sense
[? Change rejected] Give the RfC originator a good explanation for the rejection

7 Release & Deployment Management and Validation & Testing Management

Description
Release & Deployment (R & D) implements IT functionality in accordance with SLAs
Validation & Testing (V & T) tests IT functionality in accordance with SLAs
Input
[R & D] Work Packages from Change Management
[V & T] Implementations from Release & Deployment
Deliverables
[R & D] Successful releases
[V & T] Errors found in releases
Metrics
[R & D] Number of releases, number of successful releases
[V & T] Number of errors found in releases
Responsible
[R & D] Release & Deployment Manager
[V & T] Validation & Testing Manager
Authority
[Release & Deployment Manager]
To manage designated personnel so as to uphold SLAs
The power to stop any Release according to Release policy
[Validation & Testing Manager]
To manage designated personnel so as to uphold SLAs
The power to stop any Release into production systems according to Release policy
[R & D, V & T personnel] The use of common sense
Process
Receive Work Package (from Change Management)
Agree on exact time for the release
Coordinate with all relevant resources
Assign responsibility for all parts of the release
Release
[Each platform from development to production]
[R & D] Implement the release
[V & T] Test the release
[? Test fails] <Release>
[R & D] Update documentation
Inform all relevant parties of the release in production
Send to <PIR>

8 Service Assets & Configuration Management

Description

Service Assets & Configuration Management (SACM) exists to mitigate risk
SACM manages a Configuration Management Database (CMDB)
SACM tracks IT assets (Configuration Items, CIs) and their relations in the CMDB

Input

Changes to CIs (from Change Management)

Deliverables

Information on relationships

Metrics

Number of updates to CMDB
Number of errors discovered in CMDB

Responsible

Service Assets & Configuration Manager

Authority

[Service Assets & Configuration Manager] To manage designated personnel so as to uphold SLAs
[Service Assets & Configuration personnel] The use of common sense

Process

N/A

9 Knowledge Management

Description

Knowledge Management (KM) exists to foster, capture and make use of relevant knowledge
- It may be more useful having KM as an organization-wide responsibility

KM tracks competence in IT and ensures IT has optimum competence
- Each employee scores his or her knowledge on all relevant services, systems and areas
  - 0 = No knowledge
  - 1 = Bare competence
  - 2 = User competence
  - 3 = Administration competence
  - 4 = Senior administrator
  - 5 = Guru

KM runs a wiki solution and inspires entering, structuring, sharing and the use of knowledge

Input

Knowledge

Deliverables

Relevant knowledge in use

Relevant competence

Metrics

Number of page views in the wiki per IT resource

Total competence score per critical system

Responsible

Knowledge Manager

Authority

[Knowledge Manager] Can demand transfer of knowledge within the organization

[Knowledge Management personnel] Can demand transfer of knowledge within organizational areas

Process

N/A

10 How to proceed

One of the main organizational challenges in adopting ITIL is making it work together with traditional hierarchical structures. The challenge arises because of responsibility gaps or overlapping responsibilities. This can be mitigated by reducing the hierarchy to a bare minimum. Then burn the existing job descriptions and replace them with clearly defined *deliverables* for every person in the organization. No manager or employee should be responsible merely for following policies or doing tasks, but rather to deliver certain outputs deemed valuable to the operation.

Those who have been wrestling with the ITIL books may appreciate the simplicity in the above processes. Those who have little familiarity with the framework may feel that the process presented is in fact a bit daunting. But fear not, there is an even simpler approach, more in line with the article mentioned in the introduction: “Processes, Automation and Human Potential”. This simplest of approaches reads like this:

```
Area definition
   Define deliverables (outputs) for the area
      Define good metrics for the deliverables
   Determine the required inputs for the deliverables
   Determine the authority needed for the area
   Assign 100% responsibility for the deliverables to the right person
      Measure the production of deliverables
      Handle low production according to "Processes, Automation and Human Potential"
   [? Sub-areas needed]
      [Each sub-area] <Area definition>
```

With this simple HyperLists , the process becomes recursive. Whenever an area would benefit from a sub-division, one goes through the same steps for the sub-areas as one does with any area of responsibility.

Notes