https://t.me/Tcs_wings_2022

Agile

The Art of Project Management

As defined by Gartner, project management is ***"the application of knowledge, skills, tools and techniques to project activities to meet the project requirements".***

Being an integral part of software engineering processes along with the business analysis and requirement specification, design, programming and testing, the project management has been a topic of considerable debate for years. Even today, when company project management practices are becoming more mature, only about half of them (54%), according to survey results by the Project Management Institute (PMI), are fully aware of the importance and value of these practices.

Regardless of industry, project management has proven to be a crucial element of a company's efficiency and its eventual success. In fact, the organizations using proven project management practices waste 28 less money and implement projects that are 2.5 times more successful.

Project management professionals conclude that the definition of a successful project is one that is not only completed on time and within budget, but one that also delivers expected benefits.

Project Management Phases

Regardless of the scope, any project should follow a sequence of actions to be controlled and managed. According to the Project Management Institute, a typical **project management process** includes the following phases:

1. Initiation;
2. Planning;
3. Execution;
4. Performance/Monitoring;
5. Project close.

Used as a roadmap to accomplish specific tasks, these phases define the project management lifecycle.

Yet, this structure is too general. A project usually has a number of internal stages within each phase. They can vary greatly depending on the scope of work, the team, the industry and the project itself.

In attempts to find a universal approach to managing any project, humanity has developed a significant number of PM techniques and methodologies.

Traditional Project Management Methodologies

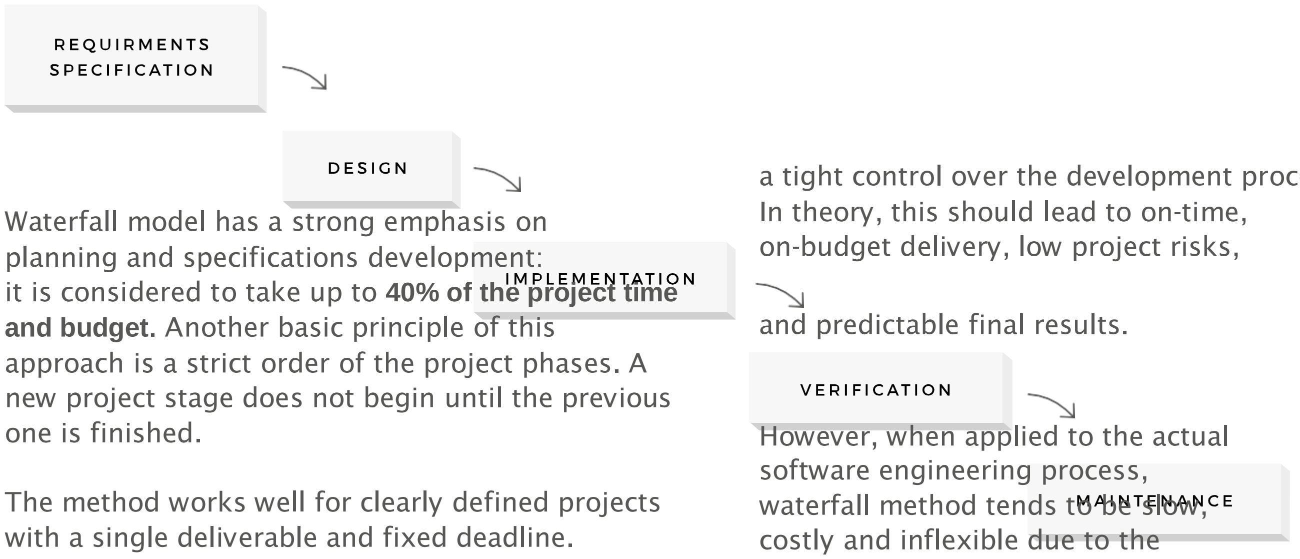
Based on the above-described classic framework, traditional methodologies take a step-by-step approach to the project execution. Thus, the project goes through the initiation, planning, execution, monitoring straight to its closure in consecutive stages.

Often called **linear**, this approach includes a num of internal phases which are sequential and executed in a chronological order.
Applied most commonly to the construction or manufacturing industry, where little or no changes are required at every stage, traditional project management has found its application in software engineering as well.

Known as a **waterfall model**, it has been a dominant software development methodology since the early 1970s, when formally described by Winston W. Royce:

> *There are two essentialsteps common to all computerprogramdevelopments,regardlessofsizeorcomplexity.Thereisfirstananalysisstep,followedsecondbyacodingstep ... This sort of very simple implementation conceptisin fact all that is required if the effort is sufficiently smalland if the final product is to be operated by those whobuilt it - as is typically donewith computer programs forinternaluse.*

Waterfall Model



Waterfall model has a strong emphasis on planning and specifications development: it is considered to take up to **40% of the project time and budget**. Another basic principle of this approach is a strict order of the project phases. A new project stage does not begin until the previous one is finished.

The method works well for clearly defined projects with a single deliverable and fixed deadline. Waterfall approach requires thorough planning, extensive project documentation and a tight control over the development process. In theory, this should lead to on-time, on-budget delivery, low project risks, and predictable final results.

However, when applied to the actual software engineering process, waterfall method tends to be slow, costly and inflexible due to the numerous restrictions. In many cases, its inability to adjust the product to the evolving market requirements often results in a huge waste of resources and eventual project failure.

Agile Project Management Methodology

As opposed to the traditional methodologies, **agile approach** has been introduced as an attempt to make software engineering flexible and efficient. With 94% of the organizations practicing agile in 2015, it has become a standard of project management.

The history of agile can be traced back to 1957: at that time Bernie Dimsdale, John von Neumann, Herb Jacobs, and Gerald Weinberg were using incremental development techniques (which are now known as Agile), building software for IBM and Motorola. Although, not knowing how to classify the approach they were practicing, they all realized clearly that it was different from the Waterfall in many ways.

However, the modern-day agile approach was officially introduced in 2001, when a group of 17 software development professionals met to discuss alternative project management methodologies. Having a clear vision of the flexible, lightweight and team-oriented software development approach, they mapped it out in the Manifesto for Agile Software Development.

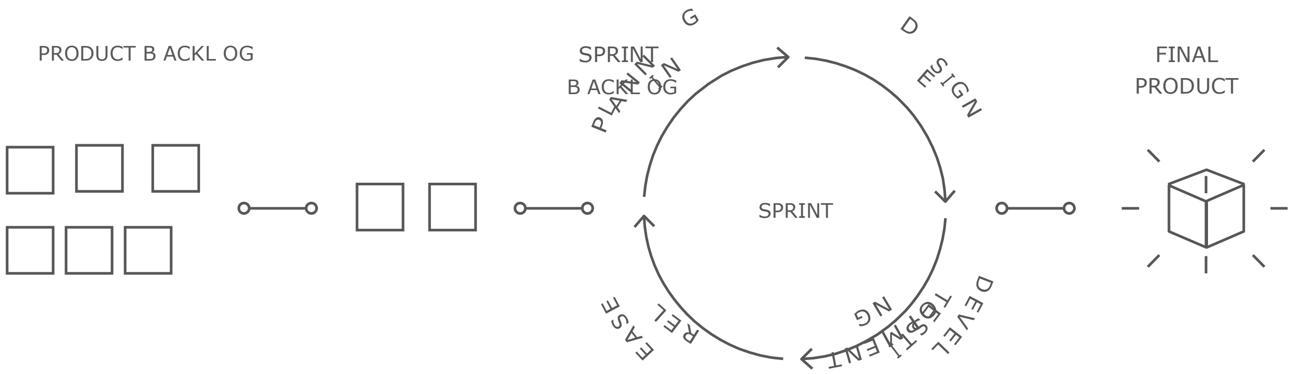
Aimed at “uncovering better ways of developing software”, the Manifesto clearly specifies the fundamental principles of the new approach:

> Throughthisworkwehavecometovalue:
>
> **Individuals and interactions**over processes and
>
> tools**Working software**over
>
> comprehensivedocumentation**Customer**
>
> **collaboration**overcontract
>
> negotiation**Respondingtochange**overfollowingaplan.

Complemented with the Twelve Principles of Agile Software, the philosophy has come to be a universal and efficient new way to manage projects.

Agile methodologies take an iterative approach to software development. Unlike a straightforward linear waterfall model, agile projects consist of a number of smaller cycles - sprints. Each one of them is a project in miniature: it has a backlog and consists of design, implementation, testing and deployment stages within the pre-defined scope of work.

Agile Development Cycle



At the end of each Sprint, a potentially shippable product increment is delivered. Thus, with every iteration new features are added to the product, which results in the gradual project growth. With the features being validated early in the development, the chances of delivering a potentially failed product are significantly lower. Let’s summarize the main Agile aspects:

Flexibility: The scope of work may change according to new requirements.

Work breakdown: The project consists of small cycles (known as Sprints in Scrum).

Value of teamwork: The team members work closely together and have a clear vision about their responsibilities.

Iterative improvements: There is frequent reassessment of the work done within a cycle to make the final product better.

Cooperation with a client: A customer is closely engaged in the development and can change the requirements or accept the team’s suggestions.

Prioritizing flexibility and rapid turnaround, the **Agile approach offers the following benefits**, according to the recent research:

- Ability to manage the changing priorities (88%)
- Increased team productivity through daily task allocation (83%)
- Better project visibility due to the simple planning system (83%)

Agile Frameworks

Agile is an umbrella term for a vast variety of frameworks and techniques, sharing the principles and values described above. Each of them has its own areas of use and distinctive features. The most popular frameworks are Scrum, Kanban, Hybrid, Lean, Bimodal, and XP. Before discussing these frameworks in more detail, let's look at their key features.

Scrum: Roles, Sprints, and Artifacts

Scrum is a dominant agile framework. It's used exclusively by 58 percent of organizations while another 18 percent of the companies combine it with other techniques. First described in 1986 by Hirotaka Takeuchi and Ikujiro Nonaka in the New Product Development Game, it was formu- lated almost a decade after.

In 1995, Ken Schwaber and Jeff Sutherland, the authors of The Scrum Guide, presented it at the OOPSLA conference. The presentation was based on the knowledge they acquired as they applied the method during the previous few years. While, Scrum was introduced far before the Agile Manifesto, it relies on Agile principles and is consistent with the values stated in that document.

Framework	Planned Mitigation
Scrum	• The entire scope of work is broken down into short development cycles — Sprints. • The Sprint’s duration is from one to four weeks. • The team should strictly follow a work plan for each Sprint. • People involved in a project have predefined roles.
Kanban	• Development is built on workflow visualization. • The current work (work in progress or WIP) is prioritized. • There are no timeboxed development cycles. • The team can change the work plan at any time.
Hybrid	• Agile and Waterfall complement each other. • Agile software development is held under Waterfall conditions (fixed deadline, forecasted budget, and thorough risk assessment).
Bimodal	• There are two separate modes of work — traditional (Mode 1) and Agile (Mode 2). • Two separate teams are working on projects with two different goals. • The Mode 1 team maintains IT system infrastructure. • The Mode 2 team delivers innovative applications. • Cross-team collaboration is important
Lean	• The framework promotes fast software development with less effort, time, and cost. • The development cycle is as short as possible. • The product delivered early is being continuously improved. • The team is independent and has a wider range of responsibilities than those in Scrum, Bimodal, and Hybrid.
XP	

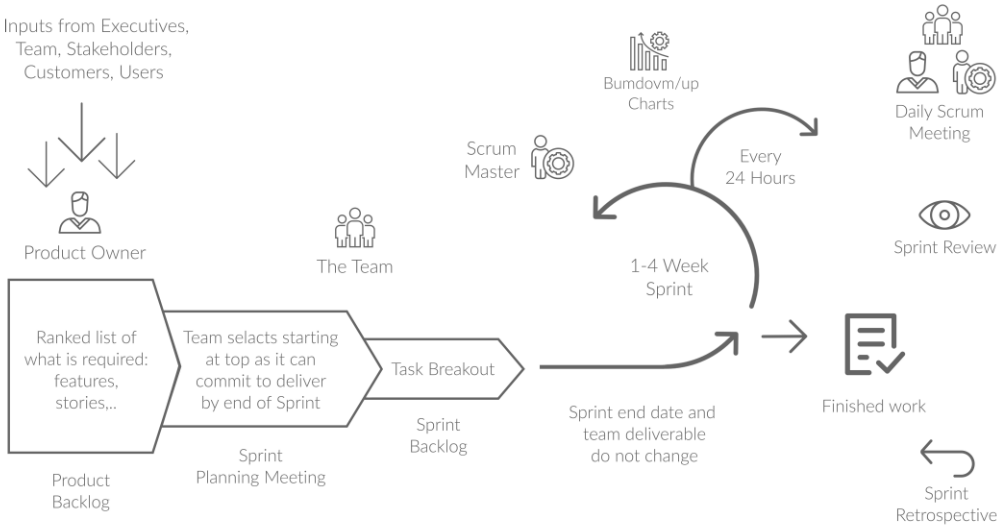
- Developers can also formulate the product’s concept

- The focus is on technical aspects of software development.
- XP introduces engineering practices aimed at helping developers write a clear code.
- Product development includes consistent stages: core writing, testing, analyzing, designing, and continuous integration of code.
- Face-to-face communication within the team and customer involvement in development are crucial

Scrum is aimed at sustaining strong collaboration between people working on complex products, and details are being changed or added.
It is based upon the systematic interactions between the three major roles: Scrum Master, Product Owner, and the Team.

- **Scrum Master** is a central figure within a project. His principal responsibility is to eliminate all the obstacles that might prevent the team from working efficiently.
- **Product Owner**, usually a customer or other stakeholder, is actively involved throughout the project, conveying the global vision of the product and providing timely feedback on the job done after every Sprint.
- **Scrum Team** is a cross-functional and self-organizing group of people that is responsible for the product implementation. It should consist of up to seven team members, in order to stay flexible and productive.

THE AGILE: SCRUM FRAMEWORK AT A GLANCE


Inputs from Executives, Team, Stakeholders, Customers, Users
Product Owner
Ranked list of what is required: features, stories,...
Product Backlog
Team selacts starting at top as it can commit to deliver by end of Sprint
Sprint Planning Meeting
The Team
Task Breakout
Sprint Backlog
Scrum Master
Bumdovm/up Charts
Every 24 Hours
1-4 Week Sprint
Sprint end date and team deliverable do not change
Daily Scrum Meeting
Sprint Review
Finished work
Sprint Retrospective

Sprints and artifacts

A basic unit of work in scrum – **Sprint** – is a short development cycle that is needed to pro- duce a shippable product increment. A Sprint usually is between one and four weeks long: More lengthy iterations lack the predictability and flexibility that are Scrum's fundamental ben- efits. Having no standard duration (as long as it is less than four weeks), all the sprints within a project should have a fixed length. This makes it easier to plan and track progress.

Scrum relies on **three main artifacts** which are used to manage the requirements and track progress – the Product Backlog, the Sprint Backlog, the Sprint Burndown Chart. The process is formalized through a number of recurring meetings, like the Daily Scrum (Standup), the Sprint Planning, the Review and Retrospective meetings.

The Product Backlog is an ordered list of feature items that might be needed in the project's final product. It is a single source of requirements. The product Backlog updates as new requirements, fixes, features, and details are being changed or added.

The Sprint Backlog is a list of tasks the team must complete to deliver an increment of functional software at the end of each Sprint. In other words, team members agree on which product items to deliver and define a plan on how to do so.

The Sprint Burndown Chart is an illustration of the work remaining in a Sprint. It helps both the team and the Scrum Master as it shows progress on a day-to-day basis and can predict whether the Sprint goal will be achieved on schedule.

Scrum meetings

The process is formalized through a number of recurring meetings, like the Daily Scrum (Standup), the Sprint Planning, the Review, and Retrospective meetings (the Sprint Retrospective).

The Daily Scrum is a timeboxed meeting, during which a Development Team coordinates its work and sets a plan for the next 24 hours. The event lasts 15 minutes and should be held daily at the same place and time.

The work to be completed is planned at the **Sprint Planning**. Everyone involved in the Sprint (a Product Owner, a Scrum Master, and a Development Team) participates in this event. They answer two key questions: which work can be done and how this work will be done. The Sprint Planning lasts no longer than eight hours for a one-month Sprint. For shorter Sprints, the meeting usually takes less time.

The work to be completed is planned at the Sprint Planning. Everyone involved in the Sprint (a Product Owner, a Scrum Master, and a Development Team) participates in this event. They answer two key questions: which work can be done and how this work will be done. The Sprint Planning lasts no longer than eight hours for a one-month Sprint. For shorter Sprints, the meeting usually takes less time.

At the end of each Sprint, the team and the product owner meet at the Sprint Review. During this informal meeting, the team shows the work completed and answers questions about the product increment. All participants collaborate on what to do next to increase the product's value. The Sprint Review is a four-hour timeboxed meeting for one-month Sprints.

The whole team goes to **Retrospective Meetings** to reflect on their work during the Sprint. Participants discuss what went well or wrong, find ways to improve, and plan how to implement these positive changes. The Sprint Retrospective is held after the Review and before the next Sprint Planning. The event's duration is three hours for one-month Sprints.

When to use Scrum

Scrum works well for long-term, complex projects that require stakeholder feedback, which may greatly affect project requirements. So, when the exact amount of work can't be estimated, and the release date is not fixed, Scrum may be the best choice.

By setting customer needs and on-time/on- budget delivery as the highest priority, Scrum has gained the trust of 89 percent of Agile users. Thus, the list of companies using this approach is impressive. In fact, there is a public spreadsheet with such organizations, including Microsoft, IBM, Yahoo, and Google.

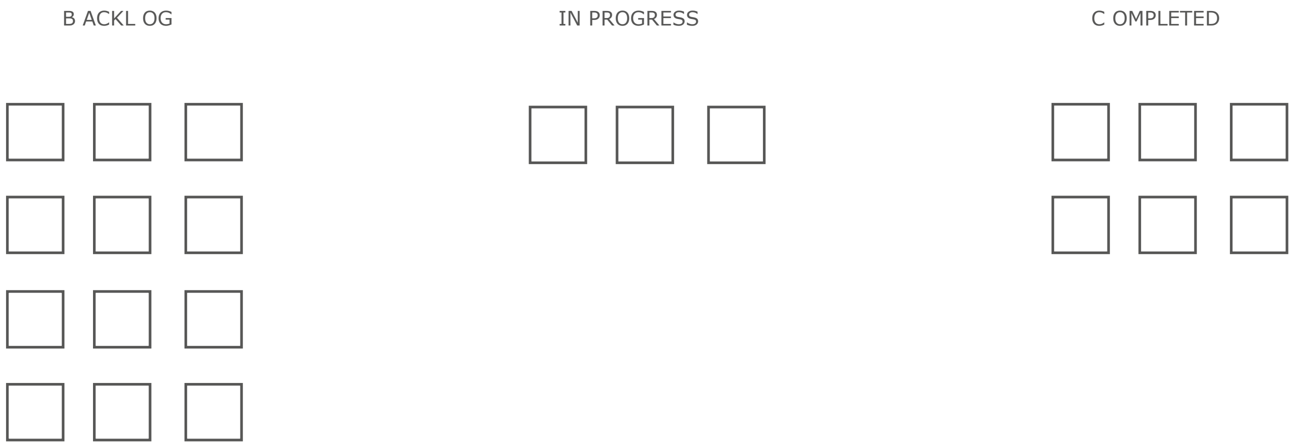
The latest research by the Scrum Alliance suggests that Scrum goes beyond IT. Companies working in the fields of finance, consulting, and entertainment choose this approach to organize their work processes and enhance cooperation with customers. In 2016, the majority of State of Scrum Report respondents (98 percent) said they are going to use this framework to move forward.

Kanban: Comprehensive Solution to Handling Work in

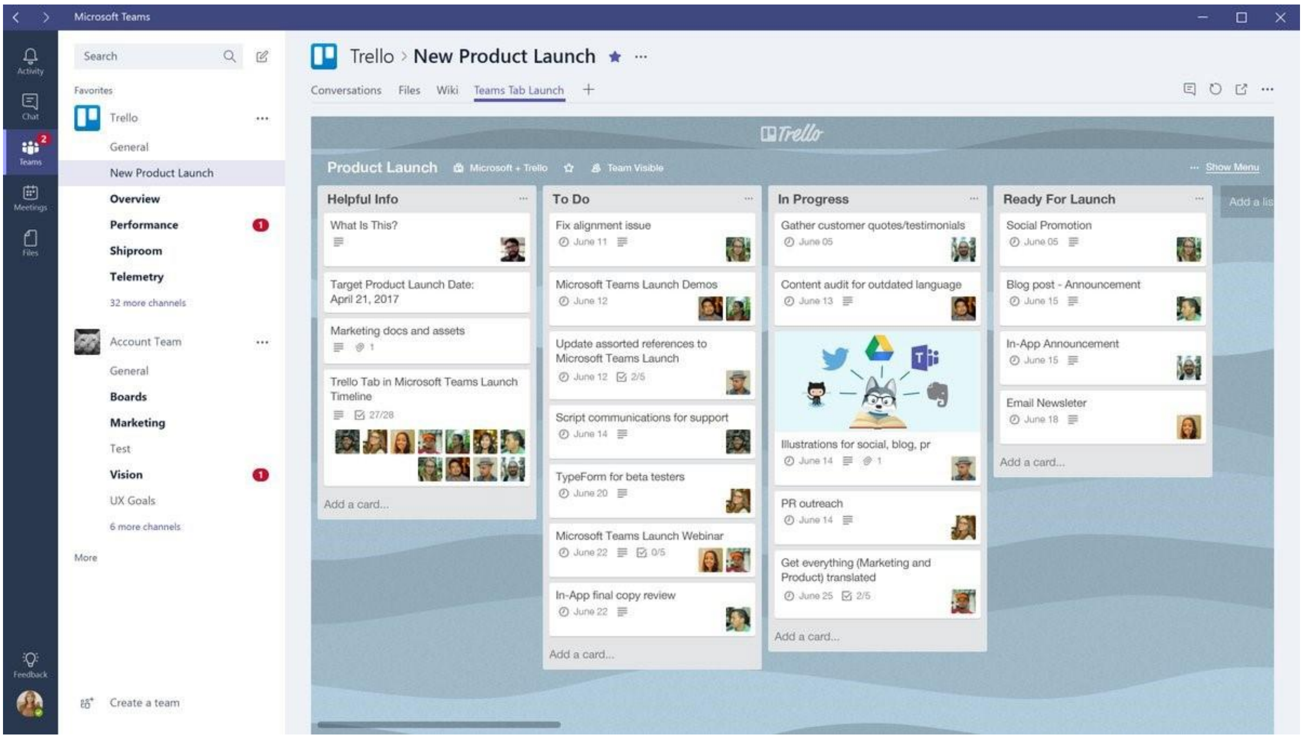
Another common project management framework is **Kanban**. Forty three percent of companies have stated that they use Kanban as one of the project management frameworks.

Originating from a visual system of cards used in Toyota manufacturing as a production control method, Kanban is simple, yet powerful, approach to developing software products. Translated as visual signal from Japanese, Kanban focuses on the visualization of the workflow and prioritizes the **work in progress (WIP)**, limiting its scope to match it effectively to the team's capacity. As soon as a task is completed, the team can take the next item from the pipeline. Thus, the development process offers more flexibility in planning, faster turnaround, clear objectives, and transparency.

Kanban Board



No standard procedures within the process, as well as the fixed iterations, are required in Kanban, as opposed to Scrum. The project development is based on the workflow visualization through a **Kanban board**, usually represented by sticky notes and whiteboards, or online tools like Trello.



Helpful Info
- What Is This?
- Target Product Launch Date: April 21, 2017
- Marketing docs and assets 1
- Trello Tab in Microsoft Teams Launch Timeline 27/28
- Add a card...

To Do
- Fix alignment issue June 11
- Microsoft Teams Launch Demos June 12
- Update assorted references to Microsoft Teams Launch June 12 2/5
- Script communications for support June 14
- TypeForm for beta testers June 20
- Microsoft Teams Launch Webinar June 22 0/5
- In-App final copy review June 22
- Add a card...

In Progress
- Gather customer quotes/testimonials June 05
- Content audit for outdated language June 13
- Illustrations for social, blog, pr June 14 1
- PR outreach June 14
- Get everything (Marketing and Product) translated June 25 2/5
- Add a card...

Ready For Launch
- Social Promotion June 05
- Blog post - Announcement June 15
- In-App Announcement June 15
- Email Newsletter June 18
- Add a card...

Trello automates and digitalizes Kanban. Due to the succinct information about a work item each Kanban card contains, everyone in the team knows who is responsible for the item, what each person's task is, when it's supposed to be finished, etc. Team members can also leave comments, attach screenshots, documents, or links to provide more details.

Teams using Kanban tools work in a cooperative manner. The ability to track progress helps coworkers understand everyone's personal input in achieving the common goal, resulting in a focus on completing the task well and on time.

When to use Kanban

Using Kanban, teams can do small releases and adapt to changing priorities. Unlike Scrum, there are no sprints with their predefined goals. Kanban is focused on doing small pieces of work as they come up. For example, if testers find errors in the product, developers try to fix them right away. Kanban, for instance, works well after the main release of the product.

Companies like Spotify and Wooga (leading mobile games development company) have been using this approach successfully over the years. Yet, 8 percent of organizations combine Scrum with Kanban techniques, using so-called Scrumban rather than the original frameworks.

Hybrid: Blend of Waterfall and Agile (Flexible Development and Thorough Project Planning)

Agile and Waterfall are two different visions of software development management. The former is about iterative development and being flexible, while the latter, promoteing step-by-step development, requires careful planning, and rejects making changes along the way.

Twenty-three percent of companies realized that using principles of both approaches can be more beneficial than choosing one of the two. The combination of the traditional Waterfall project management approach and Agile is called Hybrid.

Specialists use advantages of the Agile philosophy for software development. When it comes to budgeting, planning, and hardware set up, Waterfall works well. On the other hand, by embedding Agile practices into a traditional Waterfall work process, companies can increase chances of realizing successful projects. For example, project planning can be done in sprints, testing can be incorporated in development, and feedback can be gathered regularly. Other ways of modifying the Waterfall model include using Kanban boards and organizing retrospectives.

It should be noted that the choice of hybrid framework’s features may depend on the project. The hybrid frameworks not only imply using both approaches, depending on the project phase, but also include options to inject Agile practices into a Waterfall process.

When to use Hybrid

Hybrid is an effective solution when product delivery relies on both hardware and software operations. But, there is another reason to choose Hybrid. The situation in which a customer is not satisfied with an unspecified timeframe and budget, as well as the lack of planning, is not rare. Such uncertainty is typical for Agile. In this case, planning, requirements specification, and an application design can be accomplished in Waterfall. Agile is in place for software development and testing.

Bimodal: traditional Waterfall combined with Agile

The Bimodal approach is quite popular: It is estimated that 16 percent of companies choose it. The term "Bimodal IT" was introduced by Gartner in 2014. Bimodal is the practice of managing two separate but consistent styles of work: one focused on predictability and the other on agility.

Mode 1 is traditional; thus, it works perfectly in well-understood and predictable areas. According to Gartner, it focuses on exploiting what is known while transforming the legacy environment into a state fit for a digital world.

Mode 2 involves rapid application development. It is exploratory, nonlinear, and optimized for solving new problems. Mode 2 is especially useful for working on projects that need to be finished as quickly as possible.

Both modes require different skills, techniques, and tools. Therefore, two separate work groups are needed. These teams have two distinct goals — ensuring stability while adopting innovations. Team members focus on projects that suit their mode best.

The Mode 1 team develops and maintains applications and core systems to support long- term business needs. A company's technological capabilities depend directly on the work that's done by this team.

The Mode 2 team frequently delivers innovative applications to engage new customers and meet short-term business needs. This team may change the product's functionality after having received feedback and analyzed the market.

The teams use different delivery mechanisms and report through different organizational structures. Nevertheless, they need to communicate with each other to exchange ideas and share results.

As Sandy Kemsley specifies, Mode 2 relies on the information and services infrastructure provided by Mode 1, while Mode 1 relies on Mode 2 for testing both new product ideas and new development methods that may eventually be rolled back into Mode 1.

When to use Bimodal

If the company specializes in both long- and short-term projects that require different development and management approaches, Bimodal might be the right choice. This framework is about keeping the balance between maintaining IT system infrastructure and driving innovations. When successfully implemented, Bimodal helps organizations quickly deliver solutions that users need to stay competitive.

Lean: Eliminating Waste in Software Engineering

According to the latest estimates, 17 percent of organizations adopt **Lean**. Its popularity decreased from 2015 to 2016. Nevertheless, this framework remains one of the 5 most widely used Agile frameworks.Having the same origins as Kanban, the approach started as a technique applied to physical manufacturing. It stemmed from Toyota Production System as a management approach aimed at *"making the vehicles ordered by customers in the quickest and most efficient way, in order to deliver the vehicles as quickly as possible."*

The application of Lean principles to software development was initially introduced by Mary and Tom Poppendieck in their book Lean Software Development: An Agile Toolkit. It includes the **7 basic principles:**

- Eliminate waste
- Amplify learning and create knowledge
- Decide as late as possible
- Deliver as fast as possible
- Empower the team
- Build integrity/quality in
- See the whole

Now let's have a closer look at these principles.

Eliminating waste. In terms of a project, a term "waste" refers to anything that is not adding the value to the project and thus should be eliminated. In software engineering, this can be idle time, unnecessary features, or defects.

Amplify learning and create knowledge. In Lean, software development is perceived as an ongoing learning process. Developers don't usually write clear code on the first try. After having detected and fixed errors, they write an improved variation of the previous code. Engineers gain knowledge during development by solving problems and producing code variations. So, the best way to improve the software development environment is to amplify learning.

Decide as late as possible. Late decisions are more informed ones because they are based on facts. Keeping in mind that technologies become obsolete increasingly faster, delaying an irreversible design decision is a wise move. A major strategy for making commitments late is to reserve the capacity for the change in the system.

Deliver as fast as possible. The fourth principle is about the pros of fast software development. Short development cycles allow developers to learn more by getting feedback.

They also allow a customer to delay making a final decision about design until they know more. So, fast delivery helps eliminate waste.

Empower the team. Developers should have the right to make technical decisions as they understand the details of their work like no one else. They can create a roadmap and follow it.

Build in integrity/quality. The user's perception of the software and its characteristics must coincide. If a customer thinks that software has all the needed features and is easy to use, that system has a perceived integrity. Conceptual integrity means that the software has a coherent architecture, and scores high on usability and fitness of purpose. It can be maintained, adapted, and extended.

See the whole. Engineers should take charge of the overall efficiency of the system, instead of focusing on their small portion. If experts adhere to this principle, they can create a system with integrity.

These fundamentals perfectly describe Lean philosophy: its aim is to deliver more value through less effort, investment and time.

Lean software development is an **iterative and incremental framework**. Therefore, as in any other Agile approach, the working product increment is delivered at the early stages of development. The further progress depends largely on the product owner's feedback.

What differentiates Lean approach is that the team is not restricted to use any formal processes, such as recurring meetings or thorough task prioritization.

When to use Lean

Lean allows companies to follow a minimum viable product (MVP) development technique. It includes a deployment of a product with a minimum, sufficient set of features to satisfy early users. The idea of the MVP strategy is to gather and analyze customer feedback to know if they like this product and want to buy it. Knowledge of a customers' habits, tastes, and needs is the key to producing commercially successful products. Developers use feedback to create a roadmap for future development.

Lean works well for small, short-term projects due to their short life cycles. This approach is also appropriate if the customer can participate in a project realization as Lean requires ongoing feedback. Another important condition to the adoption of Lean is the whole team should work in one office to enable communication.

Lean works well for small, short-term projects due to their short life cycles. This approach is also appropriate if the customer can participate in a project realization as Lean requires ongoing feedback. Another important condition to the adoption of Lean is the whole team should work in one office to enable communication.

Being effectively adopted by a vast number of manufacturing companies, like Nike, Ford and Intel, Lean principles are widely used in other industries. Startups and successful companies,
e.g. Corbis, PatientKeeper, and Xerox, apply Lean software engineering practices to their processes.

Extreme Programming: Engineering Practices For Writing

Extreme Programming (XP) differs from the above-mentioned frameworks by its focus on technical aspects of software development. XP is used at 9 percent of companies.

It combines the most essential, providing agile teams with a number of tools to optimize the engineering process. Extreme Programming is a set of certain practices, applied to software engineering in order to improve its quality and ability to adapt to the changing requirements.

XP requires developers to perform a little number of engineering practices on the highest, almost extreme level possible, hence the name.

XP was introduced in the 1990s. Kent Beck, one of the initial signatories of the Agile Manifesto, invented it while working on a Chrysler Comprehensive Compensation System project. He aimed at finding ways of doing sophisticated tasks as expeditiously as possible. In 1999, he documented XP practices in the book Extreme Programming Explained: Embrace Change. The most commonly used

XP practices are:

- Test-Driven Development (TDD)
- Refactoring
- Continuous Integration
- Pair Programming

Test-Driven Development is an advanced engineering technique that uses automated unit tests to propel software design process. As opposed to the regular development cycle, where the tests are written after the code (or not written at all), TDD has a test-first approach. This means that the unit tests are written prior to the code itself.

According to this approach, the test should fail first when there is no code to accomplish the function. After that, the engineers write the code focusing on the functionality to make the test pass.

As soon as it’s done, the source code should be improved to pass all the tests. These three steps are often referred to as the RedGreen- Refactor cycle

TDD has proven to provide the following **benefits:**

1. The tests are used to capture any defects or mistakes in the code, providing constant feedback on the state of every software component. Thus, the quality of the final product is increasingly high.

2. The unit tests can be used as an always up-to-date project documentation, changing as the project evolves.

3. Being deeply involved in the product development, the team needs to be able to critically analyze it and foresee the planned outcome in order to test it properly. This keeps the team motivated and engaged, contributing to the product quality.

4. With a thorough initial testing, the debugging time is minimized.

Apart from being used within the TDD cycle, **code refactoring** is a common practice in agile software development. Basically, it’s a process of a constant code improvement through simplification and clarification. The process is solely technical and does not call for any changes in software behavior.

Extending the source code with each iteration, agile teams use refactoring as a way to weed out code clutter and duplications. This helps prevent software rot, keeping the code easy to maintain and extend.

Continuous Integration (CI) is another practice agile teams rely on for managing shared code and software testing.

We believe CI is an evolutionary development of the Agile principles.Instead of doing short iterations, developers can commit newly written parts of a code several times a day.
This way, they constantly deliver value to users.

To verify the quality of the software — through testing — and automate its deployment, teams usually use Tools like CruiseControl, Atlassian Bamboo, TeamCity or Jenkins.

In addition, CI helps maintain the shared code, eliminating the integration issues. Thus, the product’s mainline is robust and clean and can be rapidly deployed.

Pair Programming, orpairing, is considered to be a very controversial agile practice. This technique requires two engineers working together.

Pair Programming, orpairing, is considered to be a very controversial agile practice. This technique requires two engineers working together. While one of them is actually writing the code, the other one is actively involved as a watcher, making suggestions, and navigating through the process.

Being focused on both code and more abstract technical tasks, this team of two is expected to be more efficient, creating better software design and making fewer mistakes. Another benefit of this approach lies in spreading the project knowledge across team members.

However, this practice has often been accused of having a negative impact on the team's short-term productivity.

The research shows that each task usually requires 15-60 percent more time, which is a major drawback of the approach. Yet, there are some opinions that the extra time is easily compensated in the long term through the overall higher quality of the software.

When to use XP

XP provides tools to decrease risks while developing a new system, especially when developers must write code within strict timeframes. It's essential to know that XP practices are designed for small teams that don't exceed 12 people. One should choose this framework if sure that not only developers but also customers and managers will be able to work together on a project.

XP suggests unit testing as well. If programmers have enough experience creating functional tests, then XP can be used.

Extreme Programming offers engineering practices and ideas that help development teams adapt to ever-changing requirements. The key features of this framework are a high rate of customer engagement and short iterative cycles that don't exceed one week.
Also, XP suggests developers make the simplest design possible and prioritize tasks.

While XP can be used as an independent framework, some of its technical practices have become a part of other Agile approaches. Ten percent of companies choose the **Scrum/ XP Hybrid** framework, where XP engineering practices coexist with Scrum management approaches. For instance, Hybrid includes Scrum events and artifacts. The customer role evolves: it defines a Product Backlog and works together with a Development Team in the office until the project ends.

Conclusion

Agile is a way of thinking about how a software development can be managed.
Regardless of the exact frameworks and techniques they use, **98% companies have realized success from Agile projects**. Higher speed, flexibility, and productivity achieved through such approaches are the key drivers which motivate more and more organizations to switch to Agile.

Software engineering, being an extremely fast-paced industry, calls for flexibility and responsiveness in every aspect of project development. Agile frameworks allow for delivering cutting-edge products and cultivating innovative experiences while keeping the product in sync with the market trends and user requirements.

However, there is always a place for diversity. Depending on your business requirements and goals, you might still benefit from using the Waterfall model or the combination of the two.

RiO

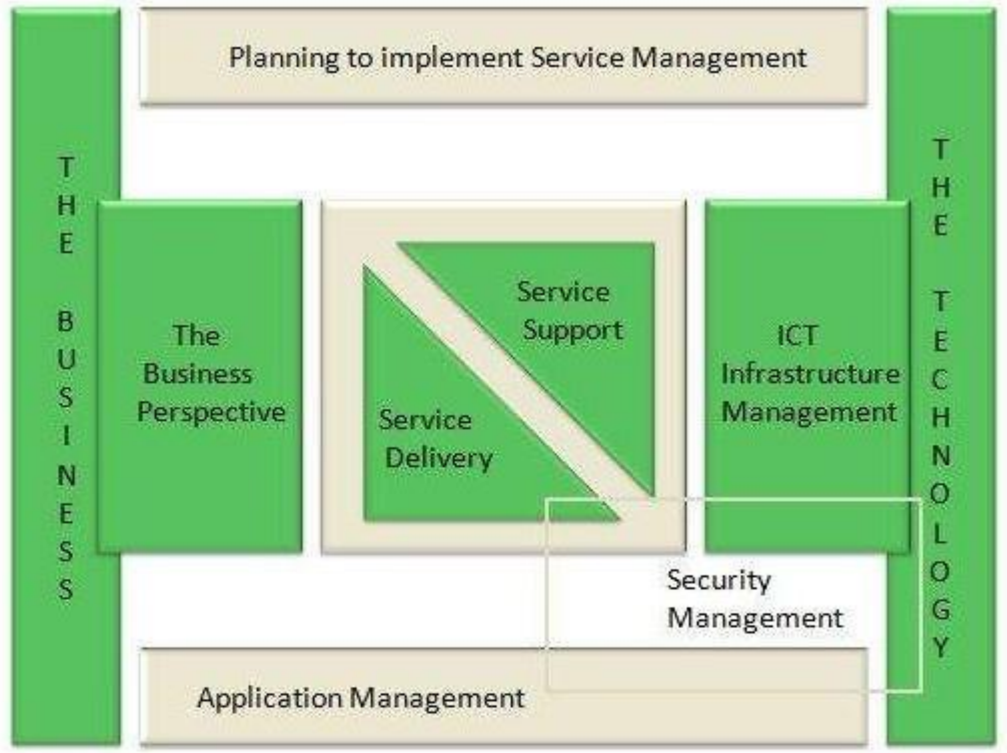
ITIL - QUICK REFERENCE GUIDE

http://www.tutorialspoint.com/itil/itil_quick_guide.htm

ITIL Overview

ITIL is a framework providing best practice guidelines on all aspects of end to end service management. It covers complete spectrum of people, processes, products and use of partners.

Now a day's ITIL is being practiced by almost every company providing IT services to the customers.



ITIL Framework

> *The processes, tasks and checklists described in ITIL are not organization-specific, but can be implemented by any organization. It gives organization a framework to plan, implement and measure IT services.*
> *ITIL was published in 1989 by **Her Majesty's Stationery Office** HMSO in **UK** on behalf of the **Central Communications and Telecommunications Agency** CCTA, now subsumed within the Office of Government Commerce* OGC.

Why ITIL is required?

ITIL helps business managers and IT managers to deliver services to the customers in effective manner and hence gaining the customer's confidence and satisfaction. Here are the areas where ITIL plays an effective role:

IT and business strategic planning

- Integrating and aligning IT and business goals Implementing
- continuous improvement
- Acquiring and retaining the right resources and skill sets
-

- Reducing costs and the Total Cost of Ownership Demonstrating
- the business value to IT
- Achieving and demonstrating Value for Money and Return on Investment. Measuring IT
- organization effectiveness and efficiency
- Developing business and IT partnerships and relationships
- Improving project delivery success
- Managing constant business and IT change

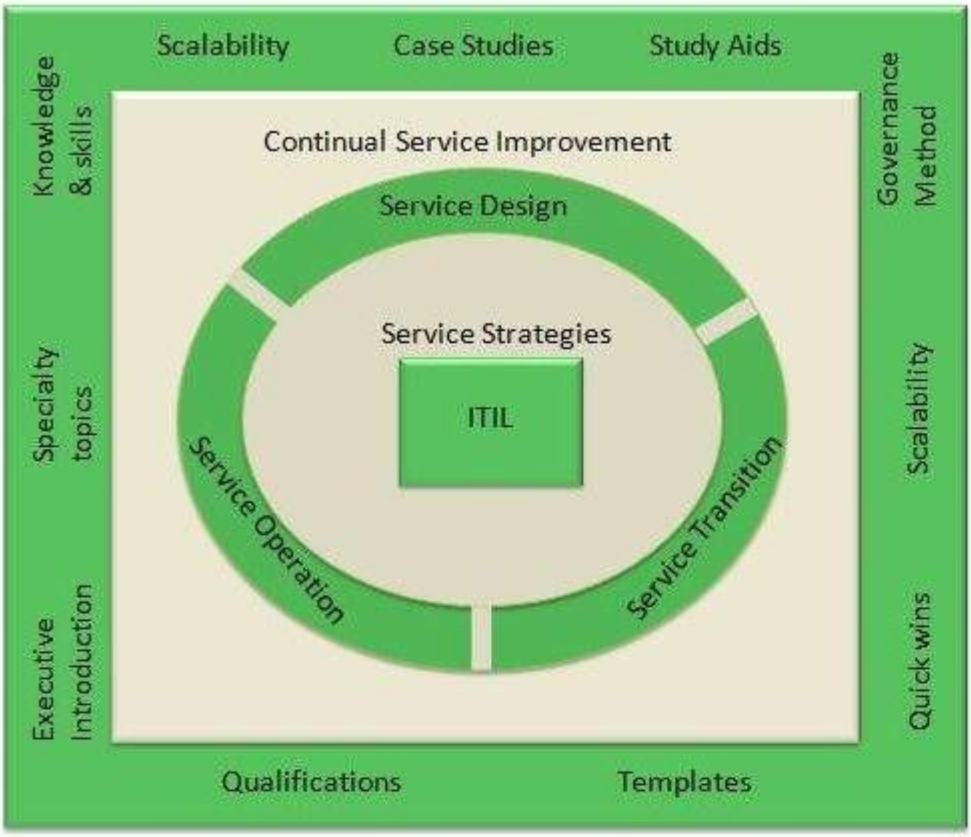
ITIL Versions

ITIL was originated as collection of books. These books of ITIL cover all aspects of IT service management. Since its origin, it has undergone many changes which lead to the following versions of ITIL:

- **ITIL V1** was the initial version of ITIL consisting of 31 books
- From 2000 to 2004, **ITIL V1** was revised and replaced by 7 books *ITILV2*. This version became globally accepted and is now used in many countries by thousands of organizations
- In 2007, ITIL v2 was modified and consolidated with 3rd version of ITIL, consisting of five core books covering the service lifecycle. ITIL V3 included **26 processes** and **4 functions**
- In 2011, the 2011 edition of V3 was published. It was an updated version released in 2007.

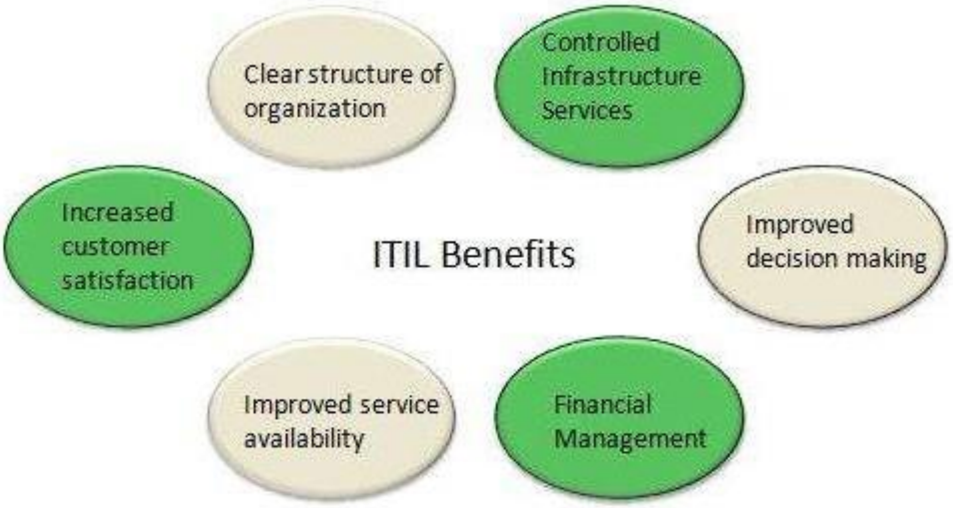
ITIL Publications

ITIL core publications include a set of five manuals: **Service Strategy, Service Design, Service Transition, Service Operation** and **Continual Service Management.**



Benefits of ITIL

Following diagram shows several benefits of ITIL:


Clear structure of organization
Controlled Infrastructure Services
Increased customer satisfaction
ITIL Benefits
Improved decision making
Improved service availability
Financial Management

ITIL V2 vs. ITIL V3

	ITIL V2	ITIL V3
Focused on product, process and people.	Focused on product, process, people and partner.	
Process oriented approach	Lifecycle based approach.	
Security management is part of evaluation	Security management is a separate process	
Emphasizes on service design and service strategy	Equal attention to all processes	
Have 10 processes and 2 functions	Have 26 processes and 4 functions.	

Service Basics

Service

Service is a means of delivering value to customers by achieving customer's desired results while working within given constraints.

Services vs. Product

	Services	Products
Services are not tangible.	Products are tangible.	
Services are produced and consumed at same time.	Products are not produced and consumed at same time.	
Services are inconsistent.	Products are consistent.	
The user participates in the production of services.	The user doesn’t participate in the production of products.	

Service Management

Service Management refers to all aspects of the management of IT service provision. According to ITIL, Service management contains all organizational capabilities for generation of added value to the customers as service.

Goals of Service Management

Here are the main goals of Service management:

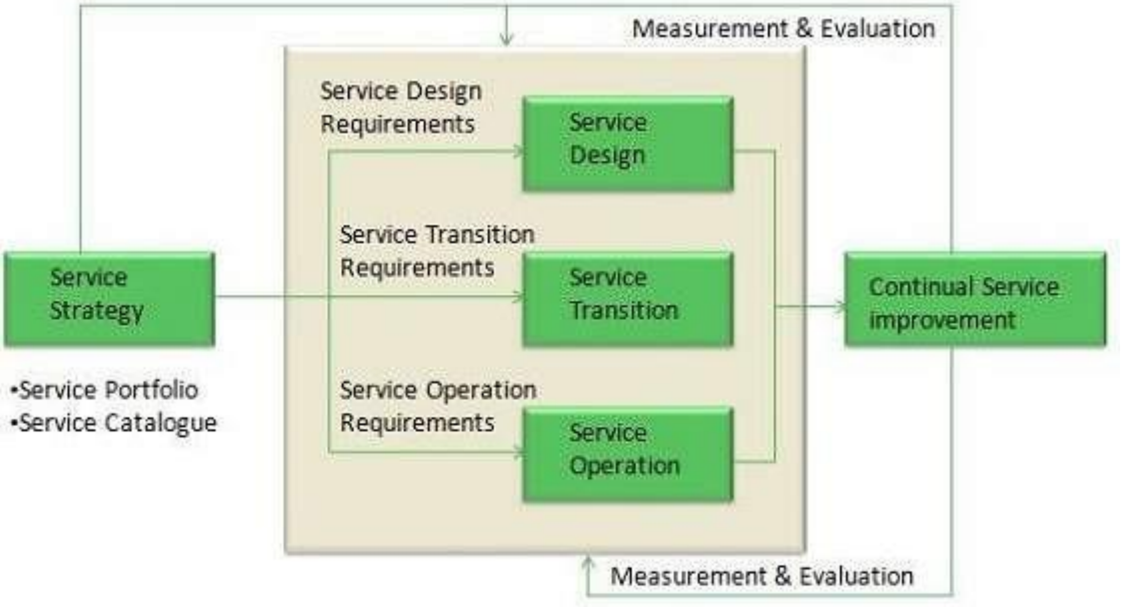
- Make IT services adaptable towards the present and future requirements of an organization and its customers
- Develop and maintain good & responsive relationship with the business Make
- effective and efficient use of all IT resources
- Optimizing the quality of delivered services Reduction of
- long term cost of service delivery

Achieving Service Management

Here are the key activities needed to be performed in order to achieve business and customer's satisfaction:

Service Lifecycle

The complete framework of ITIL is based on service lifecycle. Each lifecycle defines certain processes for effective service management.



Service lifecycle helps to amplify the service management approach and to achieve better understanding of its structure. In the coming chapters, we will be discussing all lifecycle individually with their processes and functions.

Processes and Functions

There are 26 processes and 4 functions in ITIL V3. These processes and functions are discussed in detail with each service lifecycle in further chapters.

Process

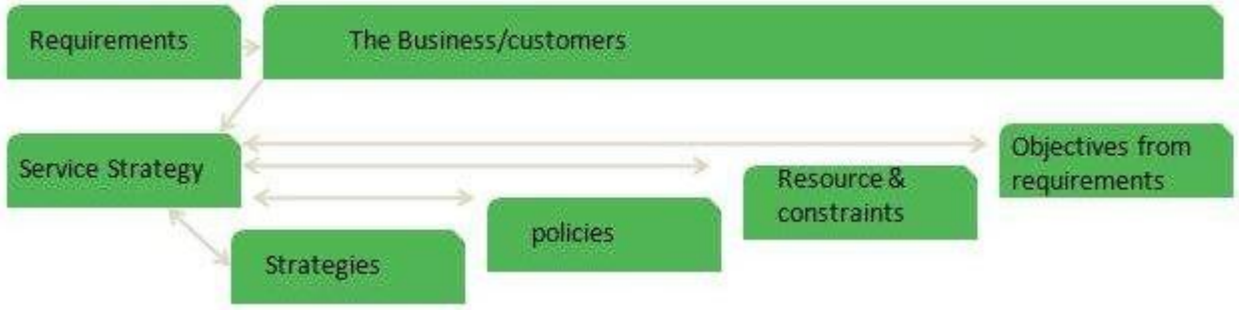
Process defines policies, standards, guidelines, activities, and work instructions. A process takes some defined inputs to produce desired results.

Functions

A function is defined as group of people or tools required to carry out one or more processes and activities.

Service Strategy Overview

Service Strategy helps to design, develop and implement service management as organizational capabilities and strategic assets as well. It enables a service provider to consistently outperform competitive alternatives over time, across business cycles, industry disruptions and changes in leadership.



Service strategy comprises of the following key concepts: Value

- creation
- Service   Assets Service
- Provider types Service
- structures
- Defining the service market
- Developing service offerings
- Financial management Service
- portfolios
- Demand management
- Return on investment

Strategic Assessment

Before crafting service strategy, a provider should first take a careful look at what it does already. The following questions can help expose a service provider's distinctive capabilities:

- Which or our services or service verities are the most distinctive? Which
- or our services or service verities are the most profitable?
- Which of our activities in our value chain or value network are the most different and effective?

Factors in Strategic Assessment

Here are the key factors that play important role in strategic assessment:

	S.N.	Description
1.	**Strengths and weaknesses** The attributes of the organization. For example resources and capabilities, service quality, skills, cost structures, product knowledge, customer relationship etc.	
2.	**Business Strategy** The perspective, position, plans and patterns are received from a business strategy.	
3.	**Critical Success factors**How will the service provider know when it is successful?	
4.	**Threats and opportunities**Includes competitive thinking. For example, is the service provider vulnerable to substitution?, or Is there a means to outperform competing alternatives?	

Value Creation

Service strategy defines a unique approach for delivering better value. According to customers service consist of two elements:

- Utility
- Warranty

Utility

Utility is perceived by the customer from the attributes of the service that have positive effect on the performance of task associated with the desired business outcomes. This is fir for purpose.

Utility is generally stated in terms of: Outcomes

- supported
- Ownership costs and risks avoided

Warranty

Warranty ensures the utility of the service is available as needed with sufficient capacity, continuity, and security. Value of warranty is communicated in terms of level of certainty.

> *Warranty is usually defined in terms of availability, capacity, continuity, and security of the utilization of the services.*

Service Assets

There are two types of service assets as listed below: Resources

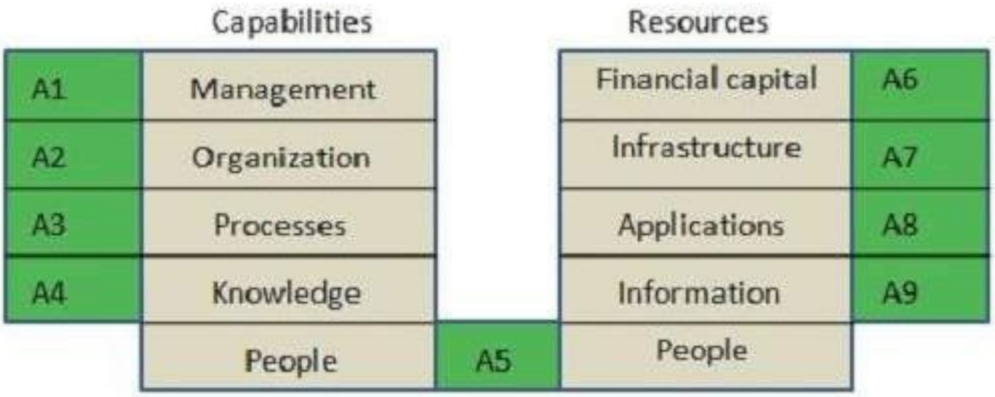
- Capabilities
-

Resources

Resources are the inputs for production. The resources are transformed by management, organization, people and knowledge.

Capabilities

Capabilities refer to skills to develop and control the resources for production. The skills are based on knowledge, experience and information.



Service Provider Types

Service Provider can be broadly classified into three types as described below:

Type I Internal Service Provider

Internal Service provider refers to the business functions within an organization. Administration, finance, human resources, and IT service providers all comes under internal service providers.

Type II Shared Service Provider

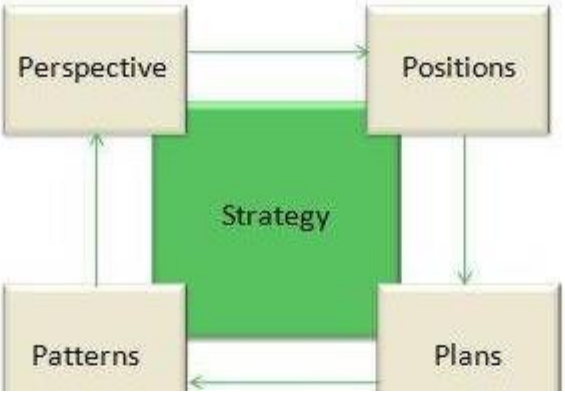
In this, business functions such as IT, human resources, and logistics are consolidated into an autonomous special unit called a Shared Service Unit *SSU*.

Type III External Service Provider

External service provider refers to the third party service providers. It can offer competitive prices and drive down unit cost by consolidating demand.

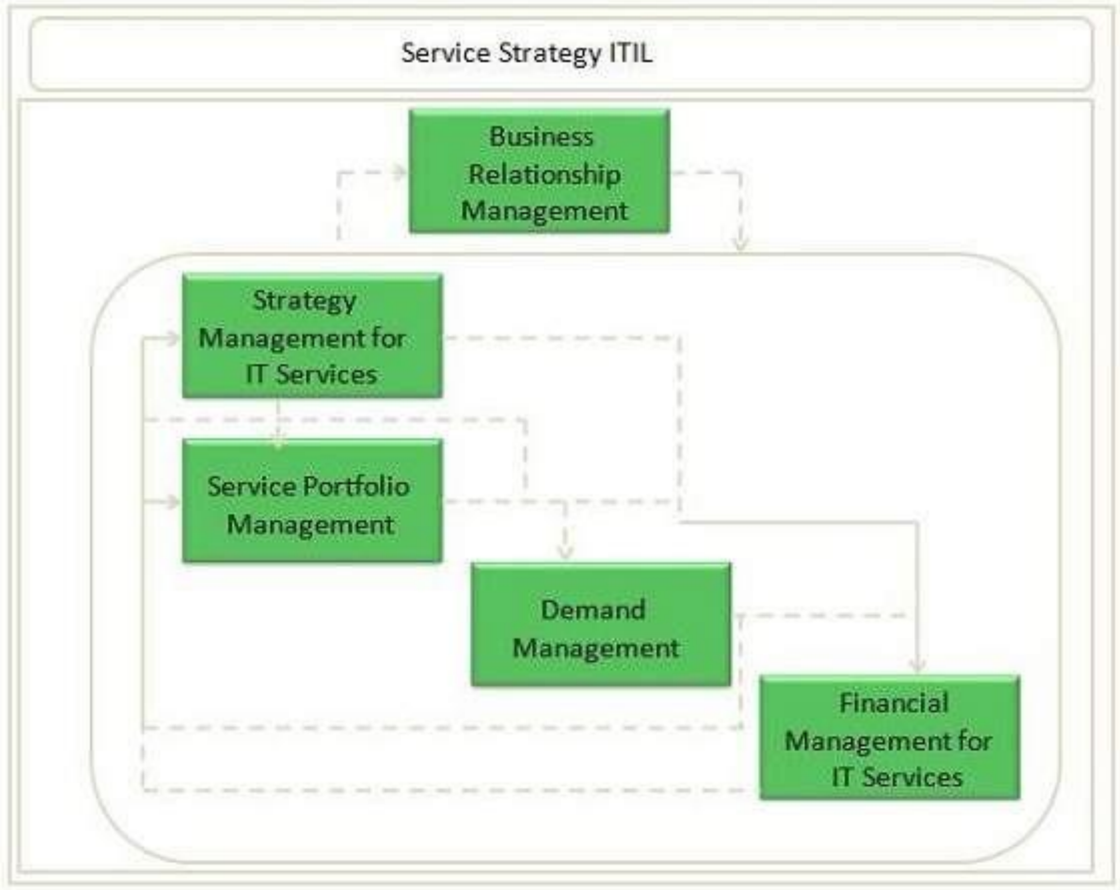
The Four Ps of strategy

The below mentioned Four Ps identify the different forms of a service strategy and are considered as entry points to service strategy.



Services strategy processes

The following diagram expresses the different processes and their relationship in service strategy:


Service Strategy ITIL
Business Relationship Management
Strategy Management for IT Services
Service Portfolio Management
Demand Management
Financial Management for IT Services

Service Strategy Roles

There are several roles that are responsible for managing different key aspects of Service Strategy. Here we will discuss all of the roles in this chapter.

S.N.	Role	Responsibility
1.	Business Relationship Manager	• • Maintains good relationship with customers Identifies • customer's needs • Ensures service provider meet customer's need Works closely with Service Level Manager
2.	Demand Manager	• Responsible for understanding, anticipating, and influencing customer demand for services • Works with capacity manager to ensure that service provider has sufficient capacity to meet the required demand
3.	Financial Manager	• Responsible for accounting, budgeting, and charging

		requirements
4.	IT Steering Group *ISG*	• Sets direction and strategy for IT services • Reviews the business and IT strategies in order to make sure that they are aligned • Sets priorities of service development programs
5.	Service Portfolio Manager	• Decides on a strategy to serve customers in cooperation with ISG • Develops service provider's offering and capabilities
6.	Service Strategy Manager	• Works with ISG in producing and maintaining the service provider's strategy • Responsible for communicating and implementing service strategy

Strategy Generation

For positive results, service provider needs to plan his services strategically. A good service strategy defines a unique approach for delivering better value.

> *Service Strategy Manager is the process owner of this process.*

Service Generation Sub processes

Strategy Management comprises of four activities as listed below: Defining the

- market
- Developing offerings
- Developing strategic assets
- Measuring and Preparation for implementation of strategy

Defining the market

It is necessary to take survey of services available in the market. It gives a clear perspective of cost and quality of services already present and what new service can be offered in competitive environment.

Developing offerings

In this service provider develops a portfolio which contain all the services that are visible and available for the customer. Service portfolio is developed in order to represent all binding service investments towards the market.

Developing strategic assets

It deals with buying new technologies, resources and capabilities to offer law-cost and high-value service to the customer.

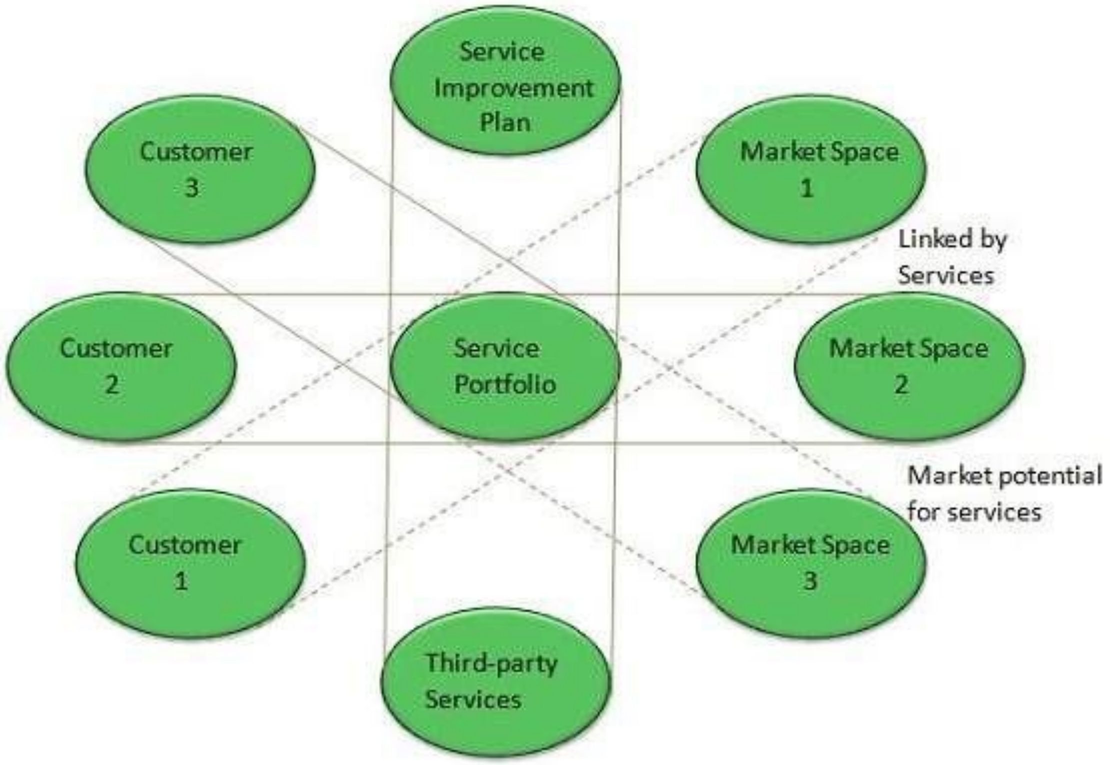
Measuring and Preparation for implementation of strategy

In order to measure success or failure of the strategy, all critical success factors are measured. Also the completion in the market is observed and priorities are

Service Portfolio Management

Service Portfolios

Service portfolio contains description of all the services engaged throughout the service lifecycle. It also represents the commitment and investment made by service provider across all customers and market spaces.



> *Service catalogue is subset of service portfolio and contains presently active services in service operation phase. We will discuss service catalogue in detail as part of service design process.*

Service portfolio management ensures that the service provider is offering right combination of services to meet the customer's need.

> *Service Portfolio Manager is the process owner of this process.*

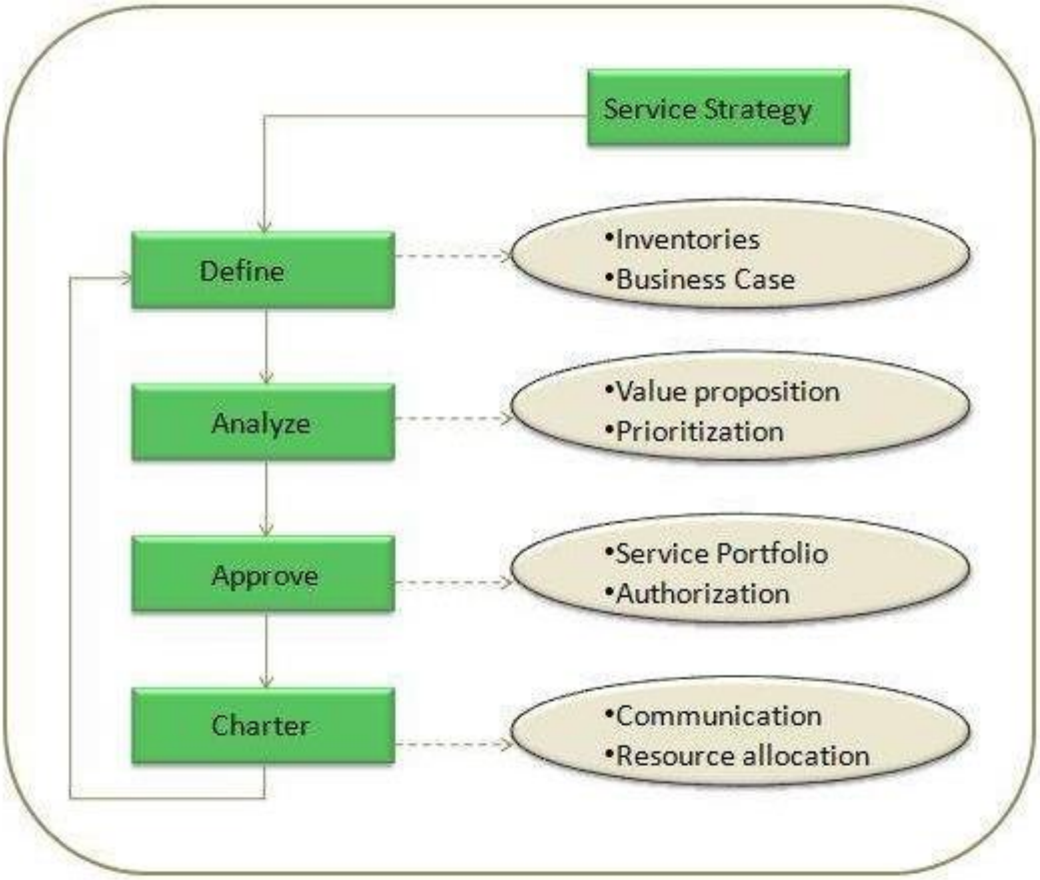
The purpose of service portfolio management is to provide answer to the following questions: Why should

- customer buy this service?
- Why should they buy from us?
- What form does the pricing structure take?
- What are our strengths and weaknesses, priorities and risks?

- How should we apply our resources and capabilities?

Sub Processes

Service portfolio management includes sub processes as shown in the following diagram:



Define

The purpose of this process is to define desired results of a service.

Analyze

The purpose of this process is to analyze the impact of proposed new service or changed service on existing services in service portfolio.

Approve

The purpose of this process is to submit change proposal to change management and to initiate the design stage for the new or changed service if change proposal is authorized.

Charter

The purpose of this process is to communicate decisions, allocate resources and charter services.

Business Relationship Management

Business Relationship Management generally includes: Managing

- personal relationships with managers Providing input to
- Service Portfolio Management

- Ensuring that IT service provider is satisfying the customer’s need

> *Business Relationship Manager is the process owner of this process.*

Sub-Processes

Business Relationship Management includes following sub-processes: Maintain

- Customer Relationship
- Identify Service Requirements
- Sign up customers to Standard Services Handle
- Customer Complaints
- Monitor Customer Complaints Customer
- Satisfaction Survey

Maintain Customer Relationship

This process ensures that service provider understands customer’s need and set up relationships with new potential customers.

Identify Service Requirements

This process ensures that service provider have complete understanding of output of a service and to decide if the customer’s need can be fulfilled using an existing service offering or if a new service needs to be created.

Sign up customers to Standard Services

This process deals with customer requirements and service level agreements.

Handle Customer Complaints

The objective of this process is to record customer’s complaints and take corrective action if required.

Monitor Customer Complaints

The objective of this process is to monitor the processing status of customer’s complaints.

Customer Satisfaction Survey

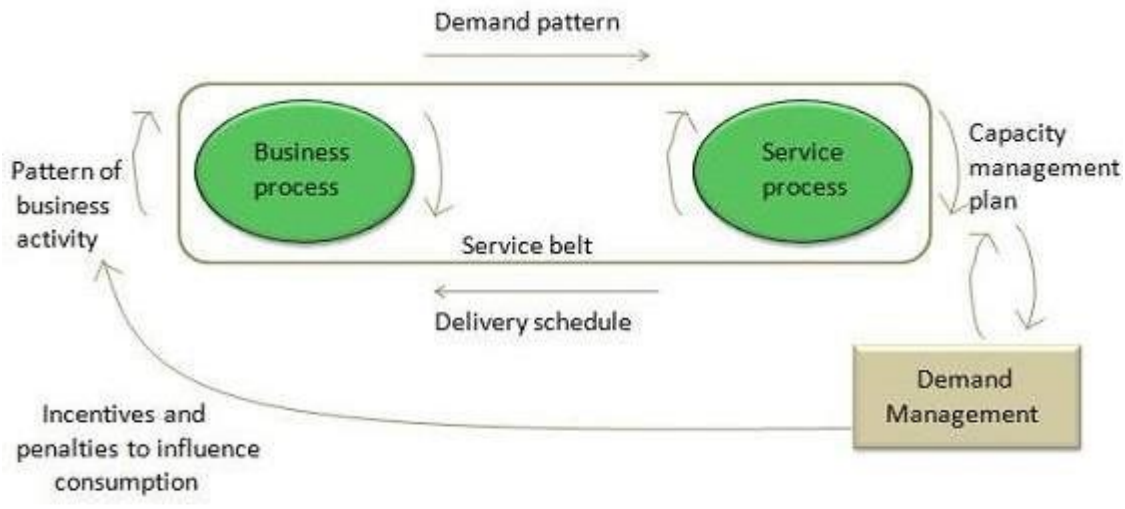
The objective of this process is to identify the scopes where customer expectations are not being met.

Demand Management

Demand Management is very important and critical process in service strategy. It helps to understand customer demand for services so that appropriate capacity can be provisioned to meet those demands.

Improper demand management leads to improper use of services and resources. Hence it’s worth to analyze customer’s demand.

> *Demand Manager is the process owner of this process.*



Stratigical Level Demand Management

Under Stratigical Demand Management we focus on two important things: Pattern of

- Business Analysis
- User Profiles

Pattern of Business Analysis *PBA*

PBA is an extremely important activity achieved by knowing customer how they operate and future requirement they might need.

User Profiles

It is the demand pattern shown by users. It can be processes, people or functions.

Tactical Level Demand Management

Under tactical level demand management we focus on Differential Charging. It is a technique to support Demand Management by charging different amounts for same IT Service Function at different times.

Challenges in Demand Management

Demand Management is critical process of service strategy. Following are the challenges that occur in this process:

- Improper analyses of customer’s demand leads to improper use of capacity. Excess capacity generates cost without creating value
- Sometimes certain amount of unused capacity is necessary to deliver service levels. Such capacity is creating value through the higher level of assurance made possible with the higher capacity
- It is required to have service level agreements, forecasting, planning, and tight coordination with the customer to reduce uncertainty in demand

- Service production cannot occur without concurrence presence of demand that consumes the output
- Arrival of demand is also influenced by demand management techniques such as off-peak pricing, volume discounts, and differentiated service levels

Financial Management

Financial Management deals with accounting, budgeting and charging activities for services. It determines all the costs of IT organization on the basis of direct and indirect costs. This process is used by all three types of service providers – internal, external or shared service providers.

> *Financial Manager is the process owner of this process.*

Benefits of Financial Management

Here are some of the benefits of Financial Management: Enhanced

- decision making
- Speed of change
- Service portfolio management
- Operational control
- Value capture and creation

Key decisions for Financial Management

Cost centre, value centre or accounting centre?

It is important to decide that how funding will be replenished. Clarity around the operating model greatly contributes to understanding the requisite, visibility of service provisioning costs, and funding is a good test of the business's confidence and perception of IT.

The IT financial cycle starts with funding applied to the resources that create output which is identified as value by the customer. This value in turn includes the funding cycle to begin again.

Chargeback: to charge or not to charge

A chargeback model provides added accountability and visibility. Charging should add value to the business.

Chargeback models vary based on simplicity of calculations and the ability for the business to understand them. Some sample chargeback model includes the following components:

Notional charges

This address whether a journal entry will be made to the corporate financial systems. Here we have two-book method in which one records costs in corporate financial systems while a second book is kept but not recorded.

This second book gives same information but reflects what would have happened if alternative method of recording had been used.

Tiered Subscription

It refers to varying levels of warranty and /or utility offered for a service, all of which have been priced, with appropriate chargeback model applied.

Metered usage

In this demand modeling is incorporated with utility computing capabilities to provide confidence in the capture of real-time usage.

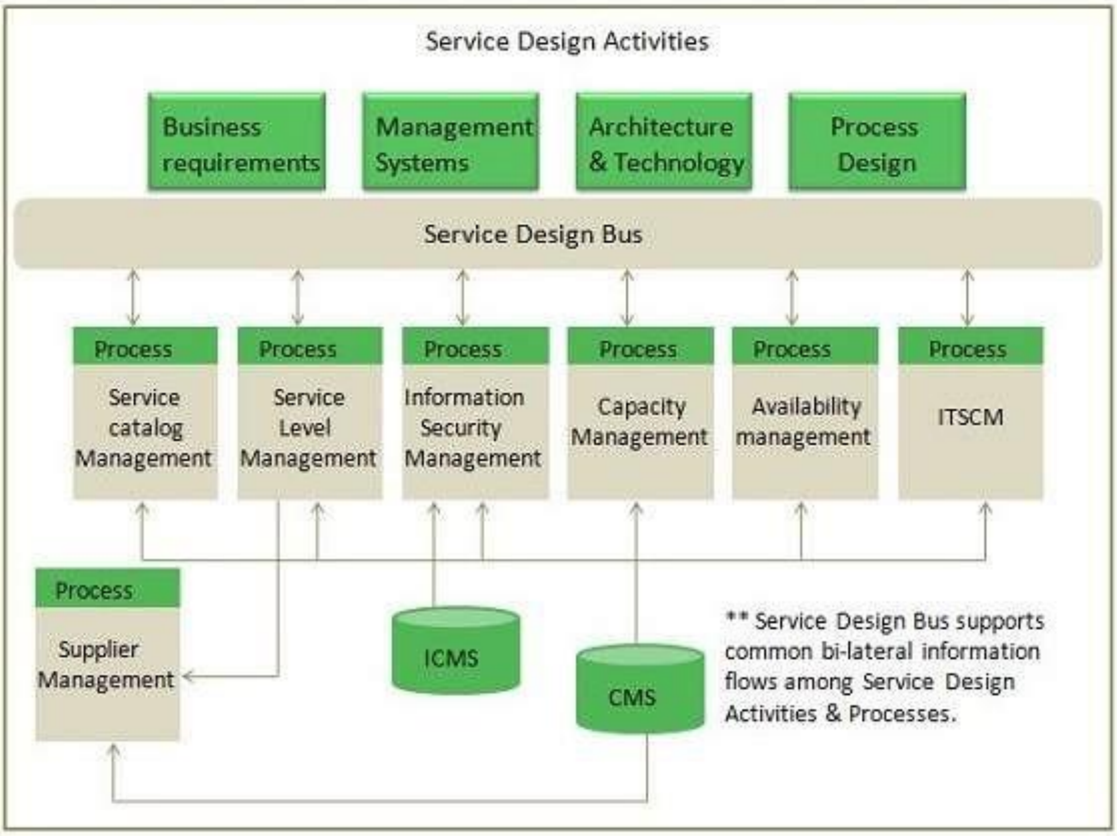
Fixed or user cost

In this cost is divided by an agreed denominator such number of users.

Service Design Overview

Service Design provides a blueprint for the services. It not only includes designing of new service but also devises changes and improvements to existing ones.

It also let the service provider know how the design capabilities for service management can be developed and acquired.


Service Design Activities
Business requirements
Management Systems
Architecture & Technology
Process Design
Service Design Bus
Process
Service catalog Management
Process
Service Level Management
Process
Information Security Management
Process
Capacity Management
Process
Availability management
Process
ITSCM
Process
Supplier Management
ICMS
CMS
** Service Design Bus supports common bi-lateral information flows among Service Design Activities & Processes.

Balanced Design

It is necessary for services to be adaptable to changing business requirements on dynamic basis. For this a balance must be maintained between following three factors:

- Functionality with the required quality.
- Resources i.e. staff, technologies, and available finances.
- Timetable

Aspects of Service Design

Service Design focuses on the following aspects:

- IT Services designed to meet business objectives.
- Services designed to be both fit for purpose and fit for use. Cost
- of ownership planed to achieve return on investment.

- Balanced functionality, cost and performance. IT
- services more stable and more predictable.
- Potential risk mitigated, so the IT service is protected from security threats.
- Design technology architectures, management architectures, and system management tools.
- Design of the measurement systems, methods, and metrics for services, processes, architectures and underlying components.
- Design of the service solution including all agreed functional requirements, resources and capabilities.

Service Oriented Architecture

Service Oriented Architecture ***SOA*** refers to developing independent usable services. SOA is defined by Organization for the **Advancement of Information Structured *OASIS*.** SOA provides more flexibility through modularity.

Prerequisites for SOA approach

Here the prerequisites required for implementation of SOA approach: Definition of

- services
- Clarity regarding interfaces and dependencies between services
- The application of standards for the development and definition of services Use of
- broadly-based technologies and tools.

Service Design Processes

Following table describes several processes in Service Design:

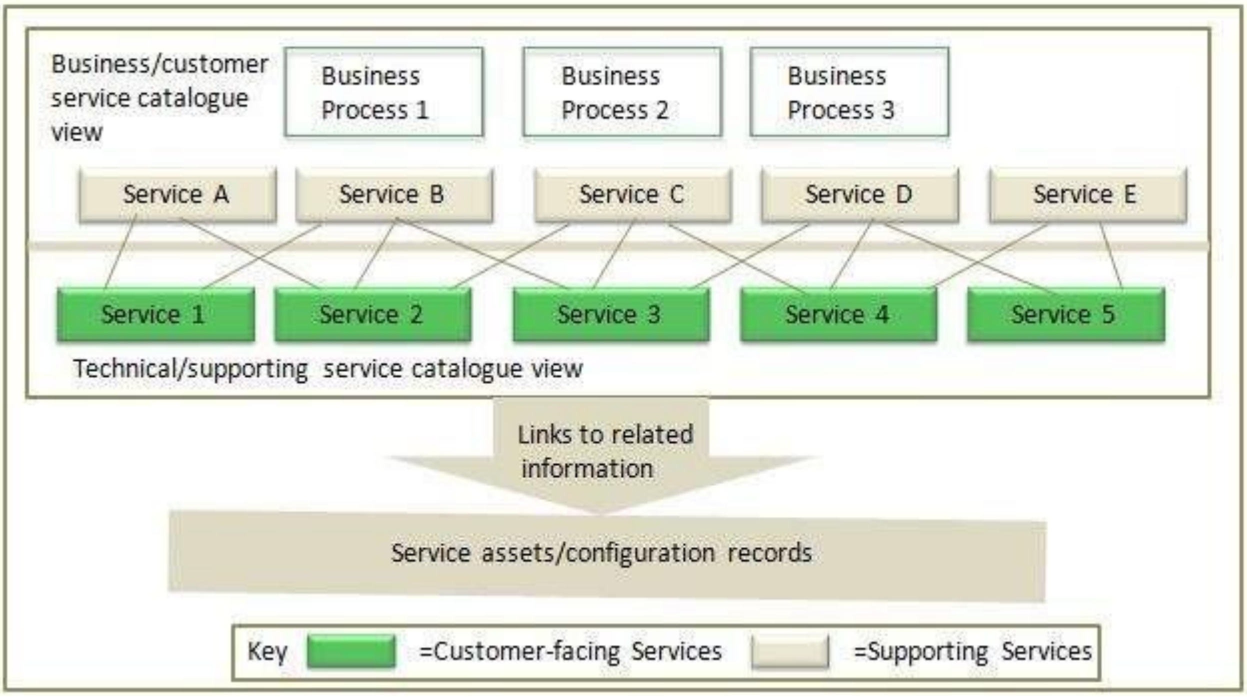
S.N.	Process Description
1.	**Design Coordination** It deals with maintaining policies, guidelines, standards, budget for service design activity.
2.	**Service Catalogue Management** This process is responsible for designing service catalogue containing service specific to the customer for which they are willing to pay.
3.	**Service Level Management** The goal of this process is to ensure quality of the services meet provisioned quality agreements.
4.	**Capacity Management** Capacity Management ensures optimal and economic usage of existing resources and future capacity requirement planning.
5.	**Availability Management** Availability Management ensures the operative services meet all agreed availability goals.
6.	**IT Service Continuity Management** This process ensures continuity of IT services regardless of any disaster occurs.
7.	**Information Security Management** This process ensures confidentiality, integrity, availability of data.

8.	**Supplier Management** This process ensures supplier relationship & performance and also ensures management of right and relevant contracts with supplier.	

Service Catalogue Management

Service Catalogue contains information of specific services for which customer are willing to pay. It is a knowledge management tool which allows employees and consultants to route their request for and about services.

> *Service catalogue is subset of service portfolio and contains presently active services in service operation phase.*


Business/customer service catalogue view
Business Process 1
Business Process 2
Business Process 3
Service A
Service B
Service C
Service D
Service E
Service 1
Service 2
Service 3
Service 4
Service 5
Technical/supporting service catalogue view
Links to related information
Service assets/configuration records
Key 🟩 =Customer-facing Services ▭ =Supporting Services

Service Catalogue Composition

Each service in the catalogue contains the following elements: A

- identification label for the service
- Description of services Related
- service request types
- Any supporting or underpinning services
- Service categorization or type that allows it to be grouped with other similar services
- Interfaces and dependencies between all services, and supporting components and configuration items *CIs* within the service catalogue and the CMS
- Clear ownership of and accountability for the service Associated
- costs
- How to request the service and how its delivery is fulfilled
- Escalation points and key contracts

- Service level agreement *SLA* data

Service Catalogue Aspects Service

catalogue has two aspects:

Business Service Catalogue

It contains all the IT services delivered to the customer, together with their relationship to the business units and the business process that rely on the IT services.

Technical Service Catalogue

It contains all the IT services delivered to the customer, together with their relationship with supporting services, shared services, components, and CIs necessary to support the provision of the service to the business.

Service catalogue management process is responsible for providing information regarding all agreed services to all authorized persons. This process also takes care of creation and maintenance of service catalogue with correct and updated information.

> *Service Catalogue Manager is the process owner of this process.*

> *Any change to service portfolio or service catalogue is subject to Change Management process.*

SCM Activities

Here are the key activities included in Service Catalogue Management: Agreeing and

- documenting a service definition with all relevant parties
- Interfacing and Service Portfolio Management to agree the contents of the Service Portfolio Management to agree the contents of the Service Portfolio and Service catalogue
- Interfacing with the business and IT Service Continuity Management on the dependencies of business units and their business processes with the supporting IT services contained within the Business Service Catalogue
- Producing and maintaining a Service Catalogue and its contents, in conjunction with service portfolio
- Interfacing with Business Relationship Management and Service Level Management to ensure that the information is aligned to the business and business process.

Service Level Management

Service Level Management *SLM* deals with negotiating, agreeing and documenting existing services with some level of policies.

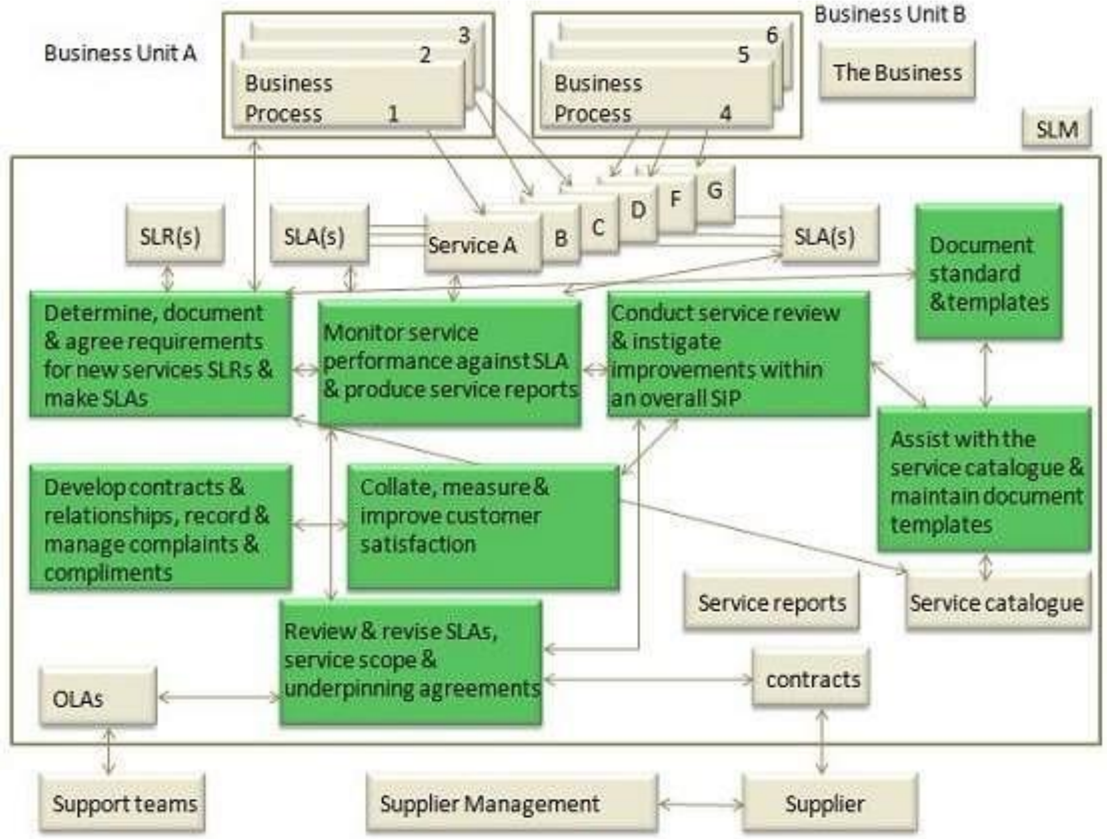
> *Service Level Manager is the process owner of this process.*

SLM deals with following two kinds of agreements:

1. Service Level Agreement *SLA*
2. Operational Level Agreement *OLA*

SLM Activities

The following diagram describes activities involved in SLM process:


Business Unit A
Business Process 1
2
3
Business Process 4
5
6
Business Unit B
The Business
SLM
SLR(s)
SLA(s)
Service A
B
C
D
F
G
SLA(s)
Document standard &templates
Determine, document & agree requirements for new services SLRs & make SLAs
Monitor service performance against SLA & produce service reports
Conduct service review & instigate improvements within an overall SIP
Assist with the service catalogue & maintain document templates
Develop contracts & relationships, record & manage complaints & compliments
Collate, measure & improve customer satisfaction
Service reports
Service catalogue
Review & revise SLAs, service scope & underpinning agreements
contracts
OLAs
Support teams
Supplier Management
Supplier

> *Service Level Requirement SLR is one of the earliest activities in Service design of Service Lifecycle. SLR needs to be drafted, once the service catalogue has been produced and SLA structure has been agreed*

Objectives

Here are the objectives of SLM:

	S.N.	Objectives
1.	Define, document, agree, monitor, measure, report, and review the level of IT service provided.	
2.	Provide and improve the relationship and communication with the business and customers.	
3.	Ensure that specific and measurable targets are developed for all IT services.	
4.	Monitor and improve customer satisfaction with the quality of service delivered	
5.	Ensure that IT and customers have a clear and unambiguous expectation of the level of service to be delivered	

Capacity Management

Capacity Management ensures proper utilization of available resources and makes future capacity requirement available in cost-effective and timely manner. Capacity Management is considered during Service Strategy and Service Design phases.

It also ensures that IT is sized in optimum and cost-effective manner by producing and regularly upgrading capacity plan.

> *Capacity Manager is the process owner of this process.*

Capacity Management Activities

The following table describes several activities involved in Capacity Management Process:

S.N.	Capacity Management Activities
1.	Producing capacity plans, enabling service provider to continue to provide services of quality defined in SLA.
2.	Assistance with identification and resolution of any incident associated with any service or component performance.
3.	Understanding customer's current and future demands for IT resources and producing forecasts for future requirements
4.	Monitoring Pattern of Business activity and service level plans through performance, utilization and throughput of IT services and the supporting infrastructure, environmental, data and applications components.
5.	Influencing demand management in conjunction with Financial Management
6.	Undertaking tuning activities to make the most efficient use of existing IT resources.
7.	Proactive improvement of service or component performance

Objectives

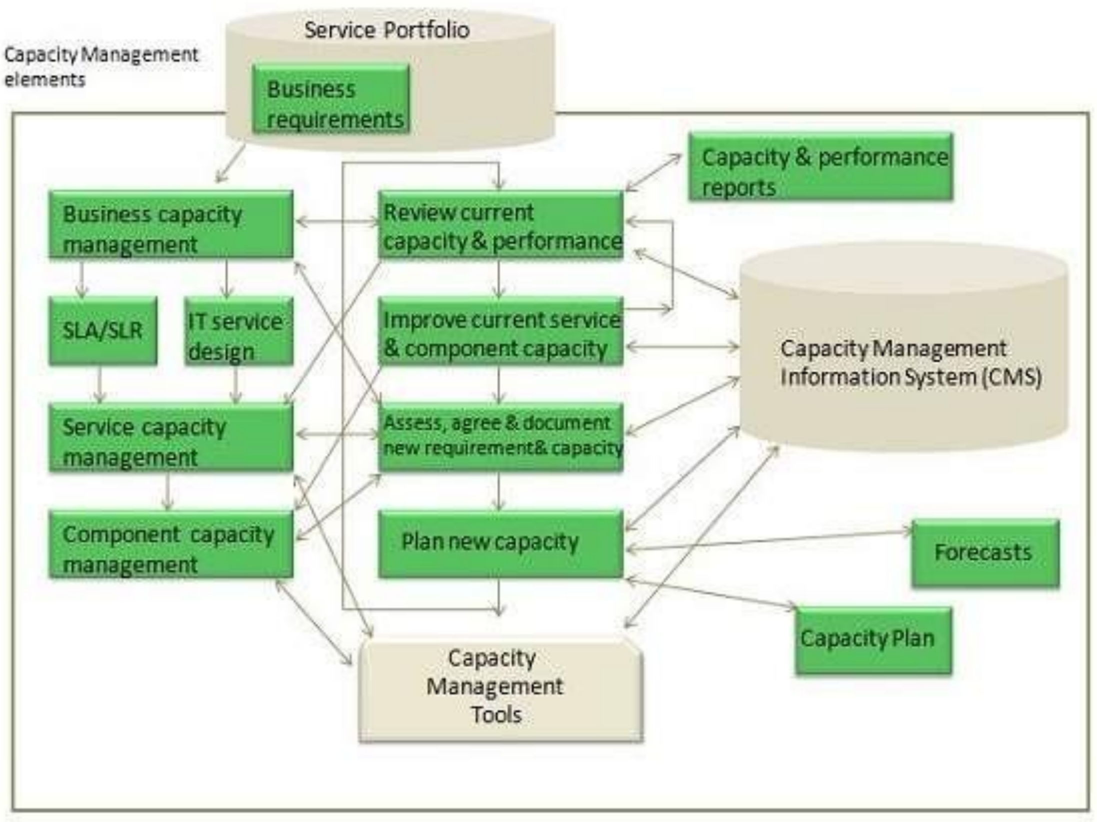
Here are the several objectives of Capacity Management:

S.N.	Objectives
1.	Produce and maintain an appropriate up-to-date capacity plan reflecting the current and future needs of the business.
2.	Provide advice and guidance to all other areas of the business and IT on all capacity and performance related issues.
3.	To manage performance and capacity of both services and resources.
4.	Assisting with diagnosis and resolution of performance and capacity related incidents and problems.
5.	Assess the impact of all changes on the capacity plan, and the performance and capacity of services and resources.
6.	Ensure that proactive measures to improve the performance of services are implemented wherever it is cost justifiable to do so.

Capacity Management Elements

Capacity Management broadly includes three components: **Business capacity management,**

Service capacity management, and **Component capacity management** as shown in the following diagram:


Capacity Management elements
Service Portfolio
Business requirements
Business capacity management
SLA/SLR
IT service design
Service capacity management
Component capacity management
Review current capacity & performance
Improve current service & component capacity
Assess, agree & document new requirement& capacity
Plan new capacity
Capacity & performance reports
Capacity Management Information System (CMS)
Forecasts
Capacity Plan
Capacity Management Tools

Availability Management

Availability Management ***AM*** ensures that IT services meet agreed availability goals. It also ensures new or changed service meet availability goals and doesn't affect the existing services.

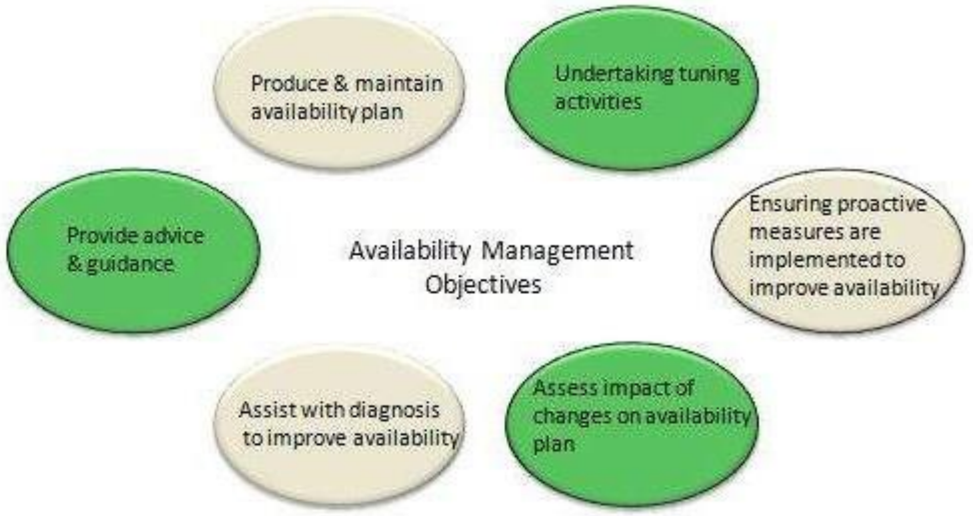
Availability is expressed as:

Agreedservicetime– Downtime**/Agreed Service Time**

> *Availability Manager is the process owner of this process.*

Objectives

Here are the objectives of Availability Management:


Produce & maintain availability plan
Undertaking tuning activities
Provide advice & guidance
Availability Management Objectives
Ensuring proactive measures are implemented to improve availability
Assist with diagnosis to improve availability
Assess impact of changes on availability plan

Availability Management Process

Availability Management process is comprises of following key elements: Reactive

- activities
- Proactive activities

Reactive activities

Activities that are involved in operational roles are known as reactive activities. Activities such as monitoring, measuring, analysis and management of all events, incidents and problem involving unavailability come under reactive activities.

Proactive activities

Activities that are involved in design and planning roles are known as proactive activities. Activities such as proactive planning, design & improvement of availability come under proactive activities.

Availability Management process is completed at following two interconnected levels: Service

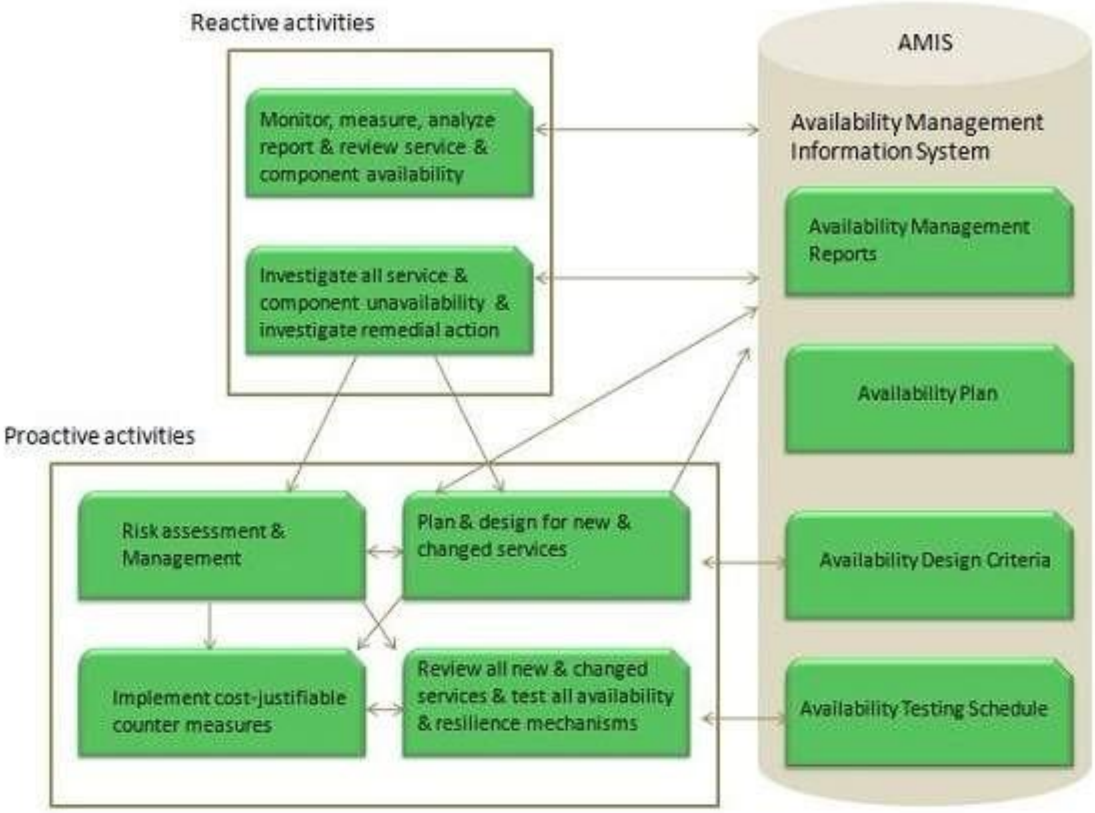
- availability
- Component availability

Service availability

It deals with availability and unavailability of service and also the impact of component availability and unavailability on service availability.

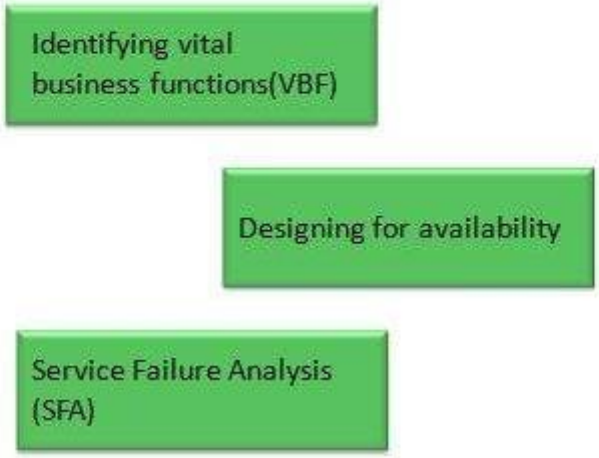
Component availability

It deals with component availability and unavailability.


Reactive activities
Monitor, measure, analyze report & review service & component availability
Investigate all service & component unavailability & investigate remedial action
AMIS
Availability Management Information System
Availability Management Reports
Availability Plan
Proactive activities
Risk assessment & Management
Plan & design for new & changed services
Availability Design Criteria
Implement cost-justifiable counter measures
Review all new & changed services & test all availability & resilience mechanisms
Availability Testing Schedule

Availability Management sub-processes

The following diagram shows sub-processes involved in Availability Management process:


Identifying vital business functions(VBF)
Designing for availability
Service Failure Analysis (SFA)

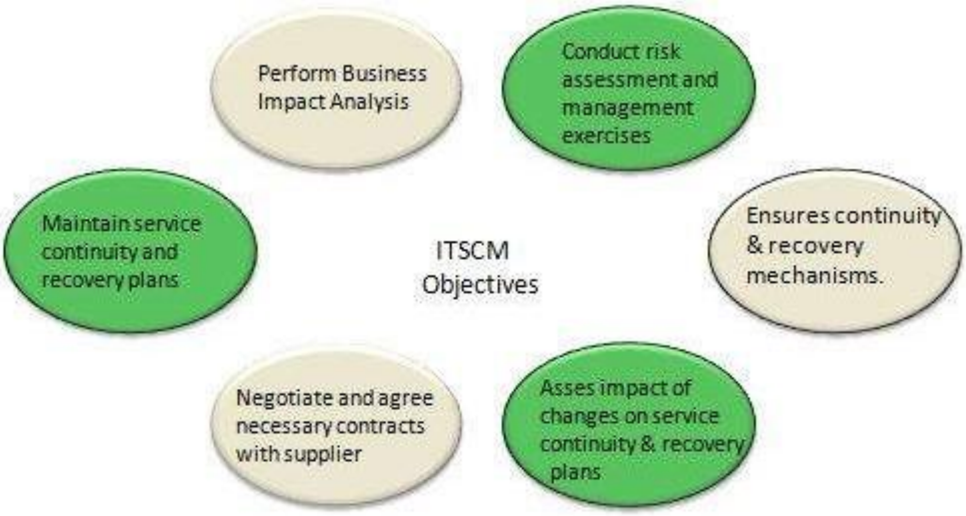
Service Continuity Management

ITSCM ensures continuity of IT service in time of any disaster. It also evaluates the level of insurance we need to protect service assets and a manuscript to recover from a disaster.

> *IT Service Continuity Manager is the process owner of this process.*

Objective

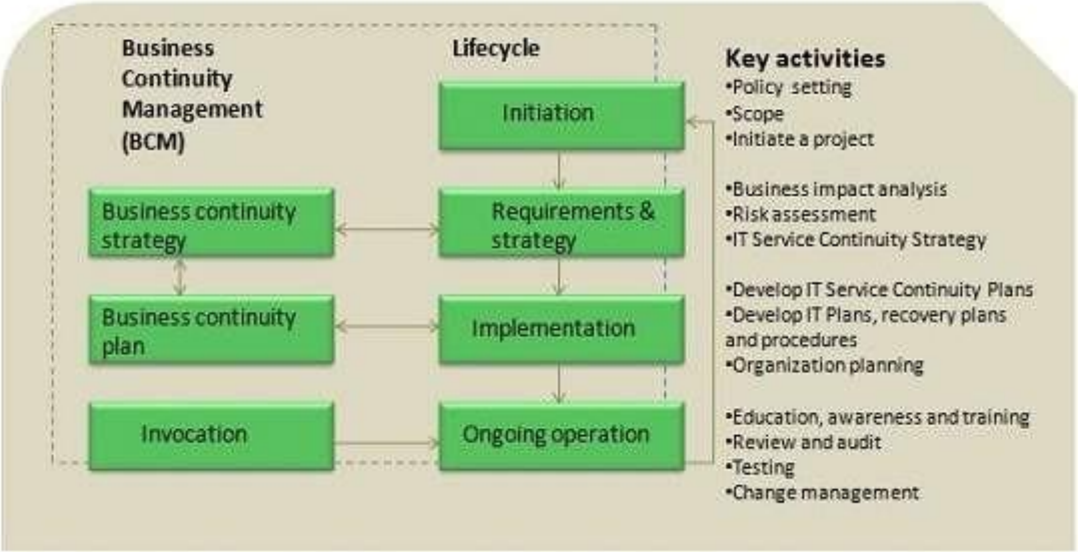
The following diagram shows the several objectives of ITSCM:


Perform Business Impact Analysis
Conduct risk assessment and management exercises
Maintain service continuity and recovery plans
ITSCM Objectives
Ensures continuity & recovery mechanisms.
Negotiate and agree necessary contracts with supplier
Asses impact of changes on service continuity & recovery plans

ITSCM Process

ITSCM process comprises of four stages: **Initiation, Requirements & strategy,**

Implementation, and **Ongoing operation.**



Information Security Management

Information Security Management *ISM* ensures confidentiality, authenticity, non-repudiation, integrity, and availability of organization data and IT services. It also ensures reasonable use of organization's information resources and appropriate management of information security risks.

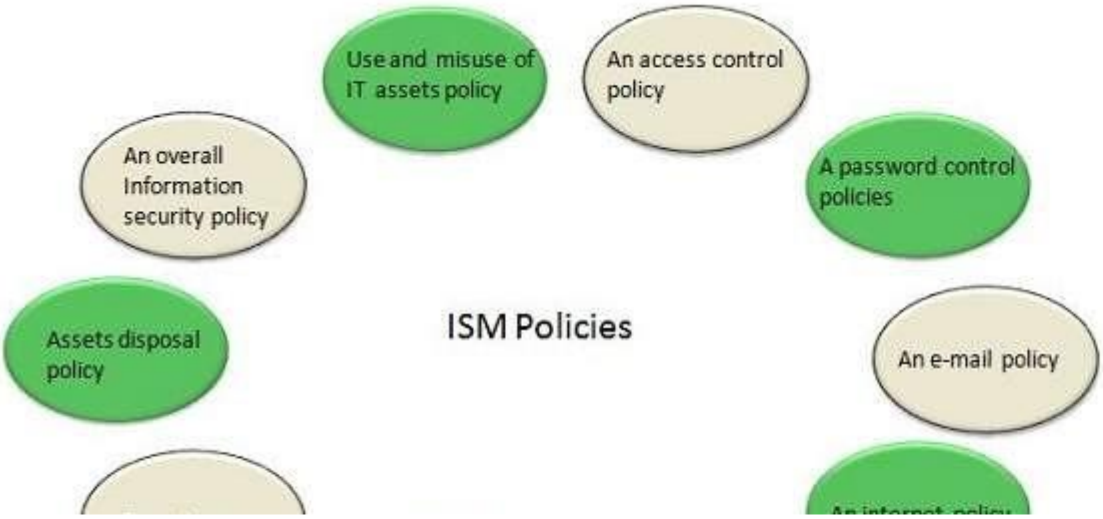
> *Information Security Manager is the process owner of this process.*

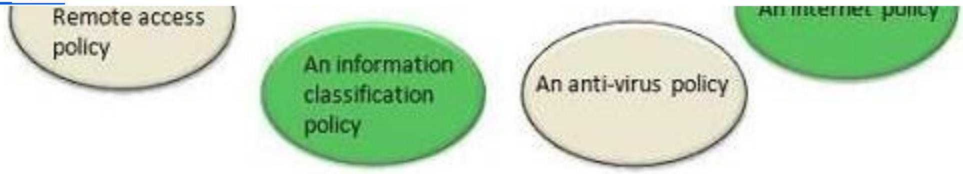
Information security is considered to be met when:

- Information is observed or disclosed on only authorized persons
- Information is complete, accurate and protected against unauthorized access *integrity*
- Information is available and usable when required, and the systems providing the information resist attack and recover from or prevent failures *availability*
- Business transaction as well information exchanges between enterprises, or with partners, can be trusted *authenticityandnon – repudiation*

ISM Security Policy

It is required for ISM security policies cover all areas of security, be appropriate, meet the needs of business and should include the policies shown in the following diagram:

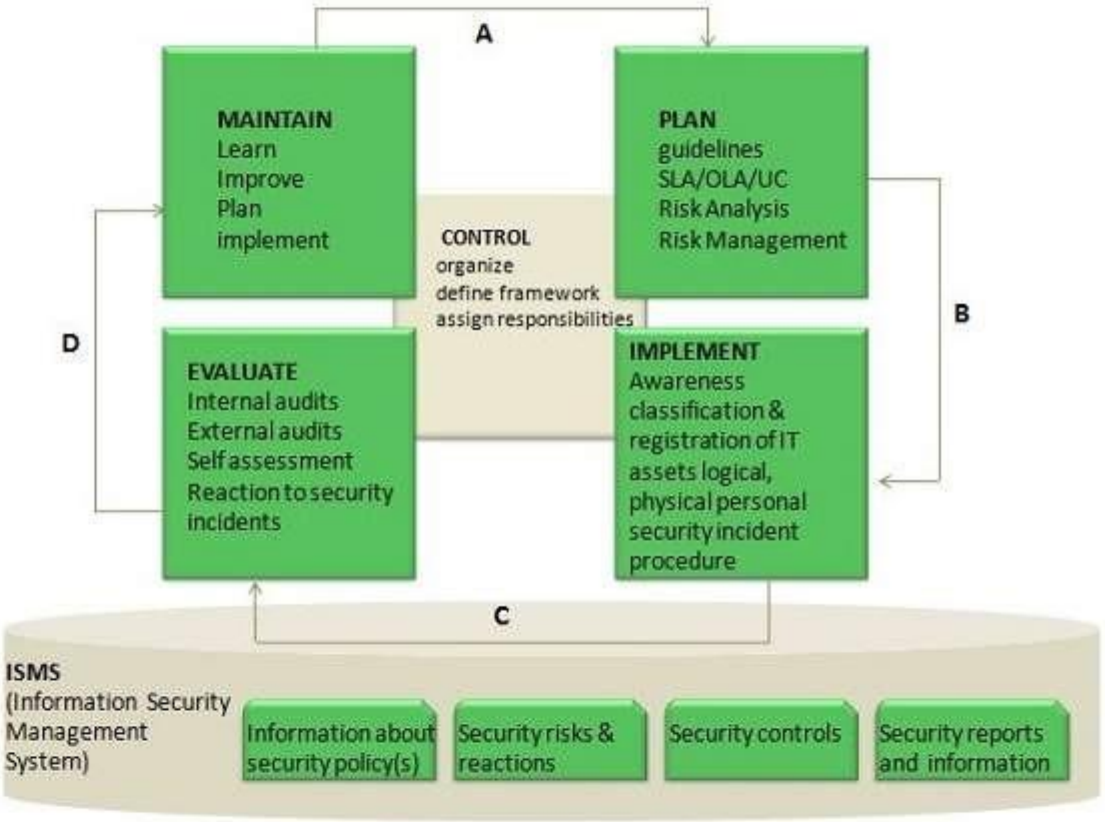

Use and misuse of IT assets policy
An access control policy
An overall Information security policy
A password control policies
Assets disposal policy
ISM Policies
An e-mail policy
An internet policy



ISM Framework

ISM Process

The following diagram shows the entire process of Information Security Management *ISM*:


A
MAINTAIN
Learn
Improve
Plan
implement
PLAN
guidelines
SLA/OLA/UC
Risk Analysis
Risk Management
CONTROL
organize
define framework
assign responsibilities
D
B
EVALUATE
Internal audits
External audits
Self assessment
Reaction to security incidents
IMPLEMENT
Awareness
classification & registration of IT assets logical, physical personal security incident procedure
C
ISMS
(Information Security Management System)
Information about security policy(s)
Security risks & reactions
Security controls
Security reports and information

Key elements in ISM Framework

ISM framework involves the following key elements:

Control

The objective of Control element is to:

- Establish an organization structure to prepare, approve and implement the information security policy
- Allocate responsibilities
- Establish and control documentation

Plan

The purpose of this element is to devise and recommend the appropriate security measures, based on an understanding of the requirements of the organization.

Implement

This key element ensures that appropriate procedures, tools and controls are in place to underpin the security policy.

Evaluation

The objective of Evaluation element is to:

- Carry out regular audits of the technical security of IT systems
- Supervise and check compliance with security policy and security requirements in SLAs and OLAs

Maintain

The objective of Maintain element is to:

- Improve on security agreements as specified in, for example, SLAs and OLAs Improve the
- implementation of security measures and controls

Preventive

This key element ensures prevention from security incidents to occur. Measures such as control of access rights, authorization, identification, and authentication and access control are required for this preventive security measures to be effective.

Reductive

It deals with minimizing any possible damage that may occur.

Detective

It is important to detect any security incident as soon as possible.

Repressive

This measure is used to counteract any repetition of security incident.

Corrective

This measure ensures damage is repaired as far as possible.

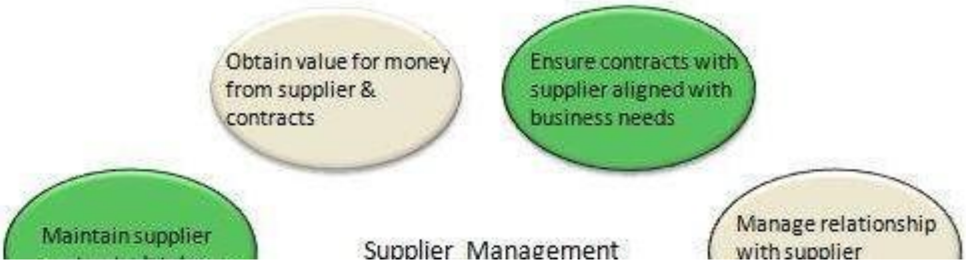
Supplier Management

Supplier Management deals with maintaining good relationship between suppliers and the partners to ensure quality IT services.

> *Supplier Manager is the process owner of this process.*

Objectives

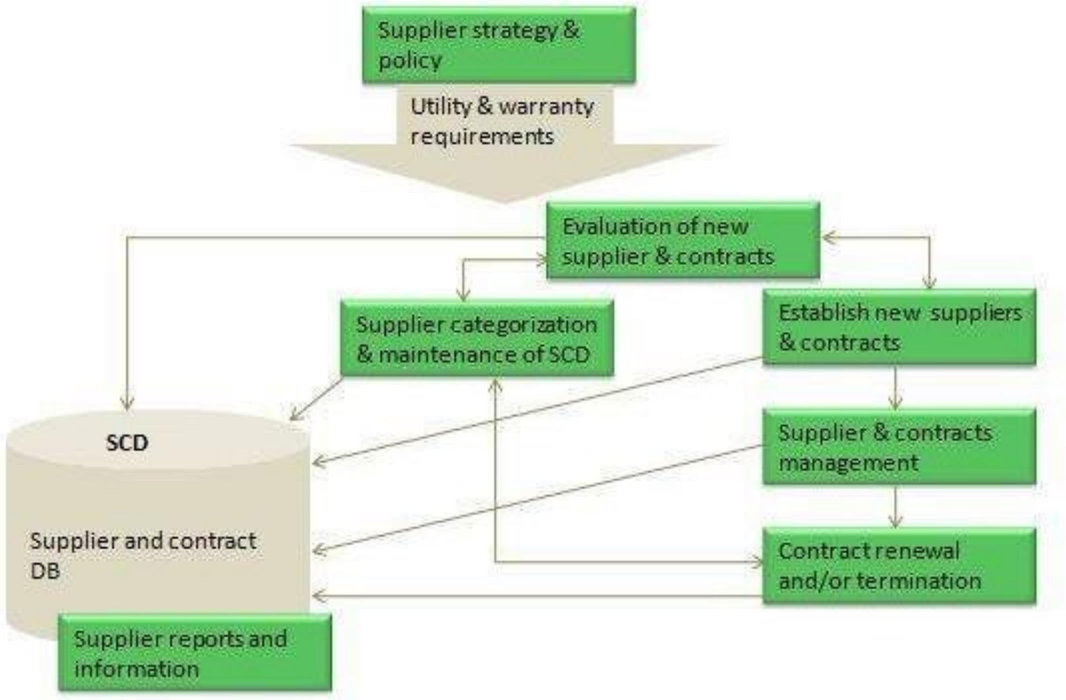
The following diagram shows several objectives of supplier management:


Obtain value for money from supplier & contracts
Ensure contracts with supplier aligned with business needs
Maintain supplier
Supplier Management
Manage relationship with supplier

contracts databases
Objectives
Negotiate and agree Contracts with supplier
Manage supplier performance

Supplier Management Process

The following diagram shows several activities included in Supplier Management:

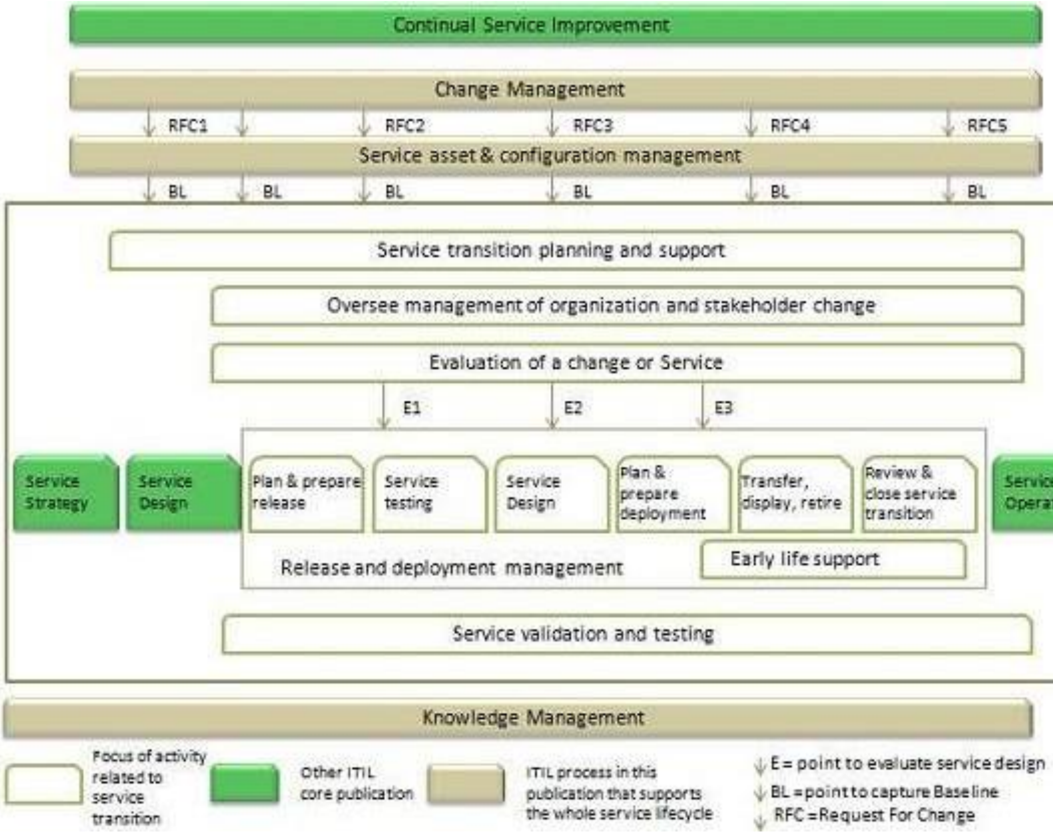

Supplier strategy & policy
Utility & warranty requirements
Evaluation of new supplier & contracts
Supplier categorization & maintenance of SCD
Establish new suppliers & contracts
SCD
Supplier and contract DB
Supplier & contracts management
Contract renewal and/or termination
Supplier reports and information

Here are the activities involved in Supplier Management Process:

- Implantation and enforcement of the supplier policy Maintenance of
- an SCD
- Supplier and contract evaluation and selection
- Supplier and contract categorization and risk assessment Contract
- review, renewal and termination
- Development, negotiation and agreement of contracts
- Maintenance of standard contracts, terms and conditions
- Management of contractual dispute resolution Management of
- sub-contracted suppliers
- Agreement and implementation of service and supplier improvement plans

Service Transition Overview

Service Transition manages transition of a new or changed service. It ensures all changes to the service management processes are carried out in coordinated way.


Continual Service Improvement
Change Management
RFC1
RFC2
RFC3
RFC4
RFC5
Service asset & configuration management
BL
Service transition planning and support
Oversee management of organization and stakeholder change
Evaluation of a change or Service
E1
E2
E3
Service Strategy
Service Design
Plan & prepare release
Service testing
Service Design
Plan & prepare deployment
Transfer, display, retire
Review & close service transition
Service Operation
Release and deployment management
Early life support
Service validation and testing
Knowledge Management
Focus of activity related to service transition
Other ITIL core publication
ITIL process in this publication that supports the whole service lifecycle
E = point to evaluate service design
BL = point to capture Baseline
RFC = Request For Change

Objectives

Here are the several objective of Service Transition:

S.N.	Objectives
1.	Provide a consistent framework for evaluating the service capability and risk profile before a new or changed service is released or deployed.
2.	Establish and maintain the integrity of all identified service assets and configurations as they evolve through the service transition stage.
3.	Plan and manage the capacity and resources required to package, build, test, and deploy a release in to production and establish the service specified in the customer and stakeholder requirements.
4.	Provide efficient repeatable build and installation mechanisms that can be used to deploy releases to the test and production environments and be rebuilt if required to restore service.
5.	Ensure that service can be managed, operated and supported in accordance with the requirements and constraints specified in service design.

Service Transition Benefits

Service Transition helps to improve several things as discussed below: Quick

- adaptability to new requirements
- Transition management of mergers, de-mergers, acquisitions and transfer of services Success rate of
- changes and releases for the business
- Success rate of changes and releases for the business
- Predictions of service levels and warranties for new and changed service

- Confidence in the degree of compliance with business and governance requirements during change
- Variation of actual against estimated and improved plans and budgets
- Productivity of business customer staff
- Timely cancellation or changes to maintenance contracts for software and hardware when components are decommissioned.
- Understanding the level of risk during and after change

Service Transition Processes

Here are the processes involved in Service Transition:

S.N.	Process Description
1.	**Transition Planning and Support** This process deals with management and control of transition plan.
2.	**Change Management** This process ensures manage and control change management process. It also prevents any unauthorized change to occur.
3.	**Service Asset and Configuration Management *SACM*** It maintains database for configuration items such as servers, switches, routers etc.
4.	**Release and Deployment Management** This process deals with management and control of movement of releases to test and live environment.
5.	**Service validation and Testing** This process deals with the quality of services offered.
6.	**Knowledge Management** This process deals with gathering, storing, analyzing, and sharing knowledge.

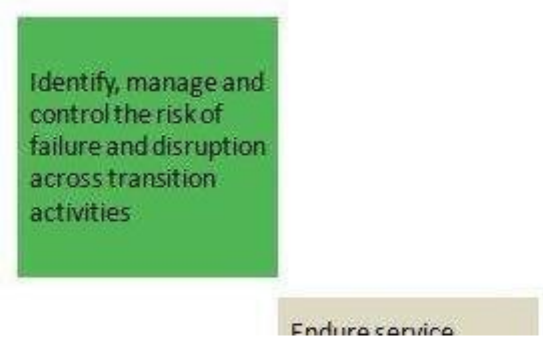
Project Management

Transition Planning and support *Projectmanagement* deals with planning the resources to deploy major release within predicted cost, time and quality estimates.

> *Project manager is the process owner of this process.*

Objectives

The following diagram describes the several objectives of project management process:


Identify, manage and control the risk of failure and disruption across transition activities
Endure service

Ensure service strategy requirements are encoded in service transition & effectively realized in service operation
Provides plans that enable the customer & business change projects to align their activities with the service transition plans

> *The Service Transition Strategy considers several aspects to organize Service Transition and allocating resources:*
>
> - *Purpose, goals, and objectives of Service Transition*
> - *Framework for Service Transition*
> - *Context, e.g. service customers, contract portfolios*
> - *Criteria*
> - *Organizations and stakeholders involved in transition People*
> - *Approach*
> - *Schedule for milestones*
> - *Financial requirements*

Planning an individual Service Transition

Service Transition plans describe tasks and activities required to release and deploy a release in to the test environment. It is good practice to deploy Service Transition plan from a proven Service Transition model.

Service Transition Plan also includes: Issues

- and risks to be managed Activities and
- tasks to be performed
- Schedules of milestones, handover and delivery dates
- Staffing, resource requirements, budgets, and timescales at each stage Lead
- times and contingency

Key Points:

- It is required to have integrated transition plans that are linked to lower level plans such as release, build and test plans.
- It is best practice to manage several releases and deployments as a programme, with each deployment run as a project.

Reviewing the Plans

It is required to verify the plans as and ask following questions before starting release or deployment:

- Have the plans been agreed and authorized by all relevant parties, e.g. customers, users, operations and support staff?
- Do the plans include the release dates, and deliverables, and refer to related change requests, known errors and problems?
- Has the service Design altered significantly such that it is no longer appropriate? Have
- potential changes in business circumstances been identified?
- Do the people who need to use it understand and have the required skills to use it?
- Have the impacts on cost, organizational, technical and commercial aspects been considered?
- Have the risk to overall services and operation capability been assessed?
- Is the service release within the SDP and scope of what the transition model addresses?

Change Management

Change in IT service refers to commissioning, decommissioning or up gradation of configuration of servers.

> *All changes are required to be implemented with minimum disruption of IT services.*

Change Management process deals with following aspects while implementing a change: Study the

- adverse impact of change and minimize it
- Create and maintain change management process Prevent
- unauthorized changes in the environment Maintain record
- of all the changes
- Post implementation review of all changes

> *Change is not implemented by change management team rather it is implemented by a technical team. Change management team only review and approve the change.*

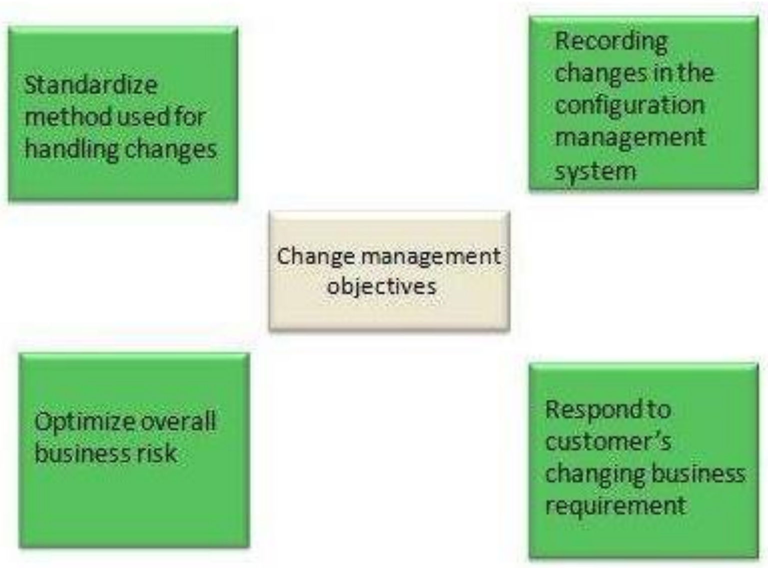
> *Change Manager is the process owner of this process.*

Key Points:

- Change manager is the person who approves the changes and closes it. He also checks whether it meets the desired result.
- Change coordinator raises changes
- Change coordinator has to send screen shots after the change in **Post Implementation Report** ***PIR***

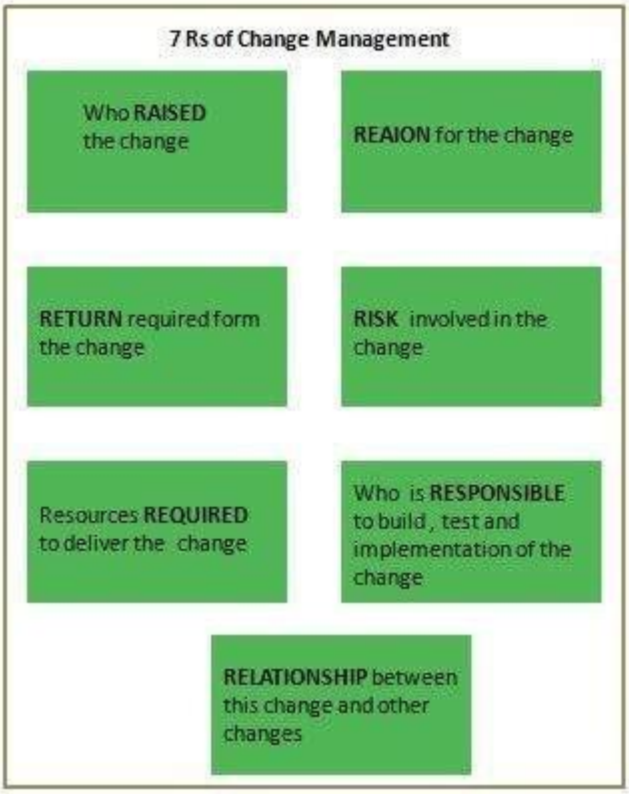
Objectives

Following are the several objectives of change management process:


Standardize method used for handling changes
Recording changes in the configuration management system
Change management objectives
Optimize overall business risk
Respond to customer's changing business requirement

Seven Rs of Change Management

These are the seven questions that must be answered for all changes. It helps to assess impact of changes and risk and benefit to the service.


7 Rs of Change Management
Who RAISED the change
REAION for the change
RETURN required form the change
RISK involved in the change
Resources REQUIRED to deliver the change
Who is RESPONSIBLE to build, test and implementation of the change
RELATIONSHIP between this change and other changes

> *Request For Change RFC is the key information source and the catalyst for the change activities of:• Create and record*
>
> - *Review*
> - *Assess and evaluate*
> - *Authorize*

> - *Plan*
> - *Coordinate*
> - *Review*
> - *Close*
>
> *Each RFC will follow a certain change model that is suitable for the nature and type of change.*

Change Models

There are basically three change models as listed below: Standard

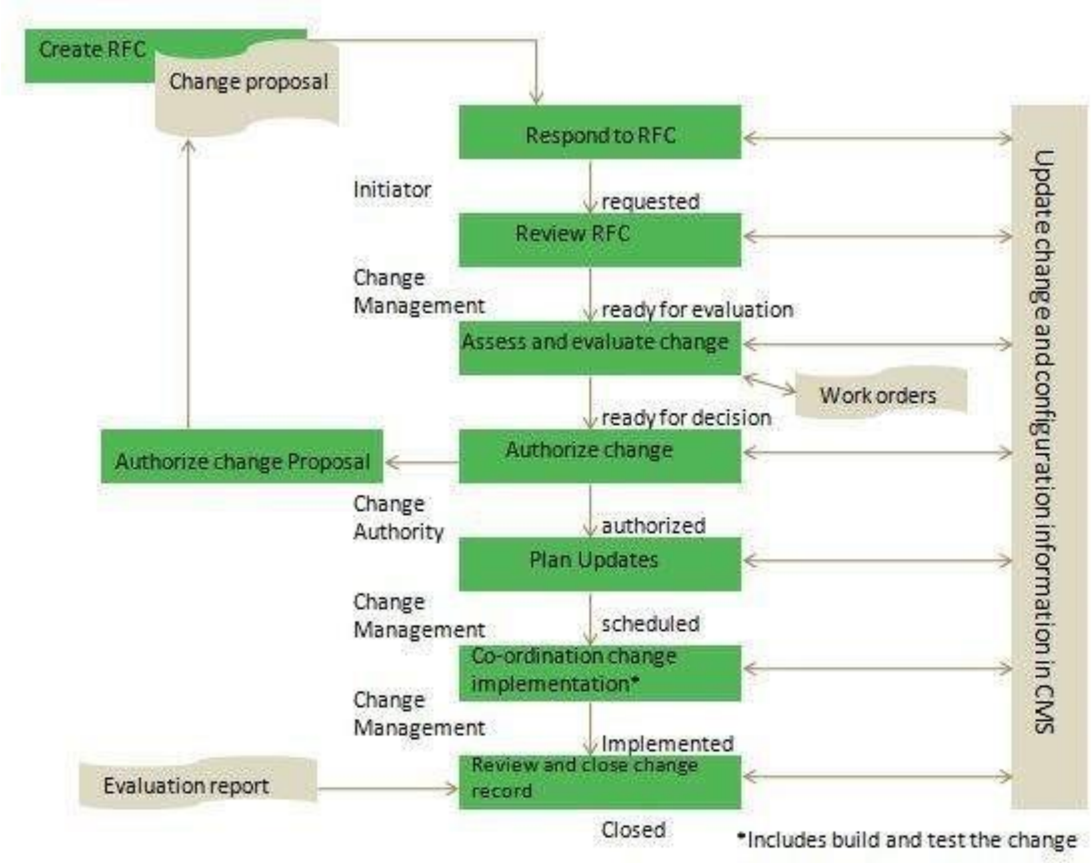
- change Model
- Normal change Model
- Emergency change Model

Standard change model

This model is used for pre-authorized repetitive, low risk and well tested changes.

Normal change model

In this model any change must go through certain steps such as assessment, authorization, and Change Advisory Board *CAB* agreement before implementation.



Emergency change model

This change model deals with highly critical changes needed to restore failed high availability service failure.

> *Change Advisory Board CAB is a body to authorize the changes and assist change management in assessing and prioritization the changes.*

Assets and Configuration Management Assets

Asset is something that has financial value with a depreciation rate attached to it. It has a cost and organization uses it for its asset value calculation.

It doesn’t have direct impact on delivering services. Anything such as servers, buildings, blackberries, switches, routers etc. comes under assets.

Configuration Items *CIs*

Configuration item is subset of service assets and have direct impact on delivering services. All servers, networks, applications that have impact on production are known as configuration item.

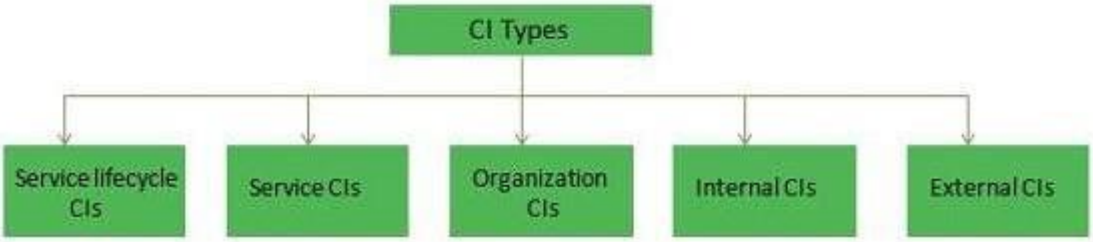
> *Building is an asset but not CI. Document is a CI not an asset.*

Hence **Service Assets and Configuration Management** ***SACM*** deals with maintaining up-to- date and verified database of all assets and CIs which are also made available to other service management processes.

> - *Configuration Manager is the process owner of this process.*
> - *SACM uses Configuration Management System CMS which contains one or more Configuration Management Databases CMDB.*

CIs Types

CIs are categorized into six as shown in the following diagram:



Service Lifecycle CIs

Service lifecycle CIs gives clear picture on: Services

- How services will be delivered? What
- benefits are expected?

- Service cost

Service CIs

Service CIs refers to:

- Service model
- Service package
- Release package
- Service resource assets Service
- capability assets

Organization CIs

Organization CIs are internal to an organization but independent of the service provider.

Internal CIs

CIs which are delivered by individual projects are known as Internal CIs.

External CIs

External CIs refer to external customer requirements and agreements, releases from supplier and external services.

Interface CIs

These CIs are required to deliver end-to-end service across a Service Provider Interface *SPI*.

Configuration Management System *CMS*

SACM uses **Configuration Management System** ***CMS*** which contains one or more **Configuration Management Databases** ***CMDB*****.** The database contains information associated with a CI such as supplier, cost, purchase date, renewal date for licenses and maintenance contracts and other related documents such as SLA etc.

There are also other attributes for Configuration Items such as: Unique

- Identifier
- CI type
- Name/description
- Supply date
- Location
- Status
- License details
- Related document masters
- Related software masters
- Historical data

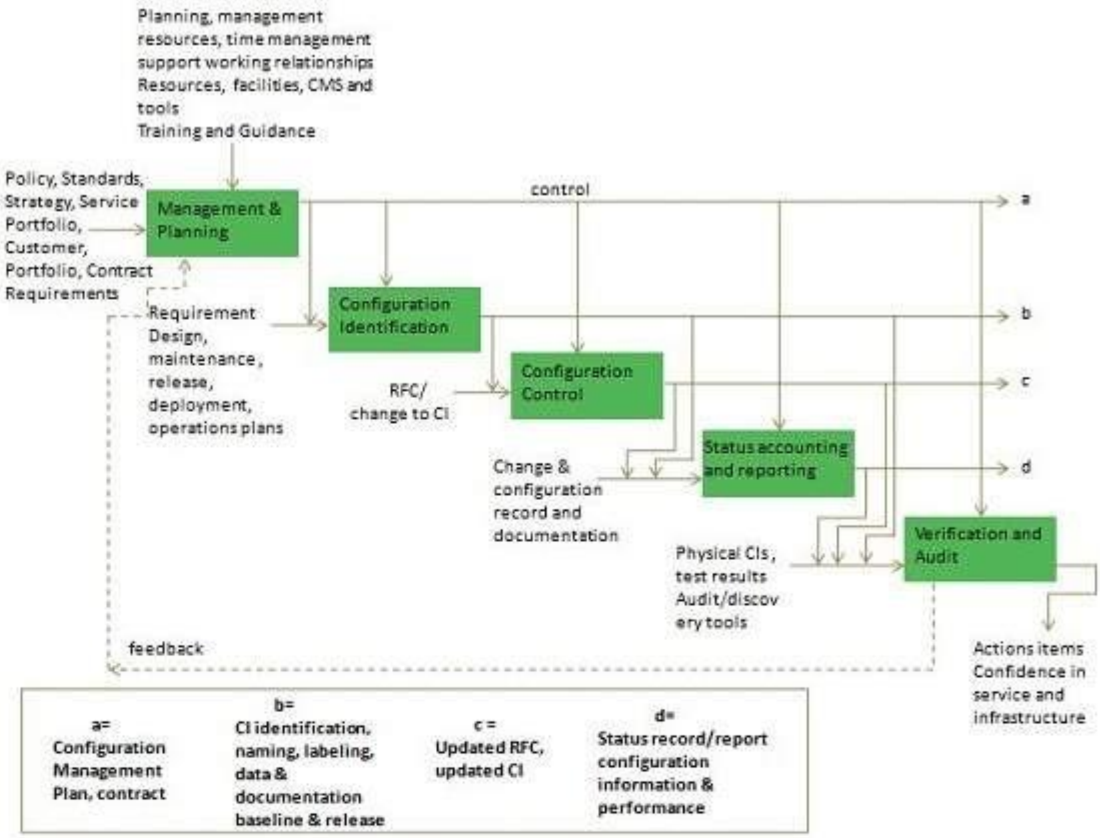
- Relationship type
- Application SLA

Service Assets and Configuration Process SACM

process comprises of following five activities:

1. Management and Planning
2. Configuration identification
3. Configuration control
4. Status accounting and reporting
5. Verification and Audit

All of the above mentioned activities are described in the following diagram:


Planning, management resources, time management support working relationships Resources, facilities, CMS and tools Training and Guidance
Policy, Standards, Strategy, Service Portfolio, Customer, Portfolio, Contract Requirements
Management & Planning
control
a
Requirement Design, maintenance, release, deployment, operations plans
Configuration Identification
b
RFC/ change to CI
Configuration Control
c
Change & configuration record and documentation
Status accounting and reporting
d
Physical CIs, test results Audit/discovery tools
Verification and Audit
Actions items Confidence in service and infrastructure
feedback
a= Configuration Management Plan, contract
b= CI identification, naming, labeling, data & documentation baseline & release
c = Updated RFC, updated CI
d= Status record/report configuration information & performance

Release and Deployment Management

Release and Deployment Management includes planning, designing, building, testing and deploying new software and hardware components in the live environment. It is important to maintain integrity of live environment by deploying correct releases.

> *Release Manager is the process owner of this process.*

Key Points:

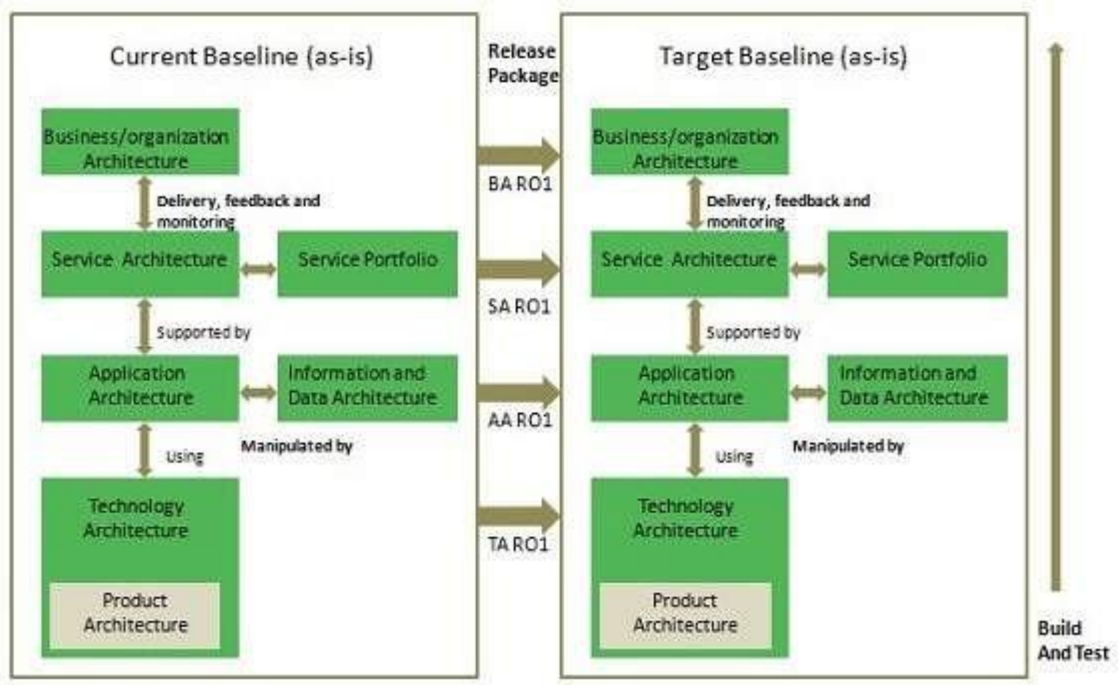
- Release Unit refers to portion of IT infrastructure that should be released together.
- In Release package each of the software in the package depends on other software in the group for its performance.

Objectives

Release and Deployment management ensures that:

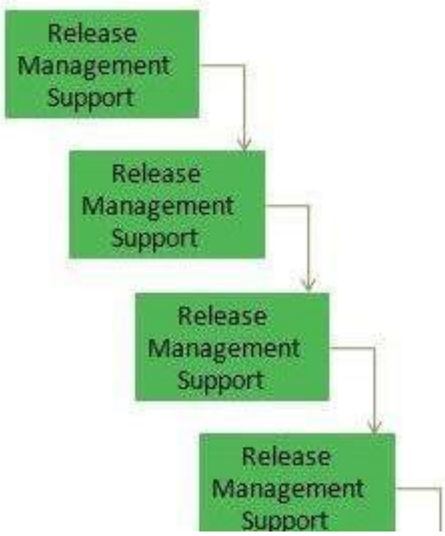
- Delivering change faster and at optimum cost and minimized risk Successful and
- on schedule deployment of release package.
- New or changed services are capable of delivering the agreed service requirements.
- There is knowledge transfer to enable the customers and users to optimize their use of service to support their business activities.
- Minimal unpredicted impact on the production services, operation and support organization.
- Customers, users and service management staff are satisfied with service transition practices and outputs.

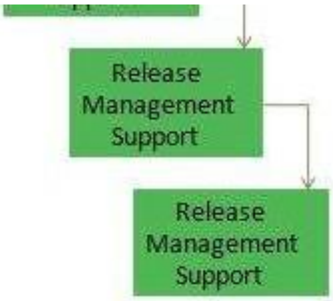
> *It is important to define appropriate release package type for a given type of release. The following diagram illustrates an example of a release package:*



Release and Deployment Management Process

Here the activities included in Release and Deployment Management process:


Release Management Support
Release Management Support
Release Management Support
Release Management Support



Release Management Support

It provides guidelines and support for the deployment of releases.

Release Planning

The objective of this process is to assign authorized changes to release packages. It also defines the scope of releases.

Release Build

This process deals with building releases and ensures all components are ready to enter the testing phase.

Release Deployment

The objective of this process is to deploy new release in the live environment and also arrange training for end users and operating staff.

Early Life Support

The purpose of this process is to resolve operational issues during initial period after release deployment.

Release Closure

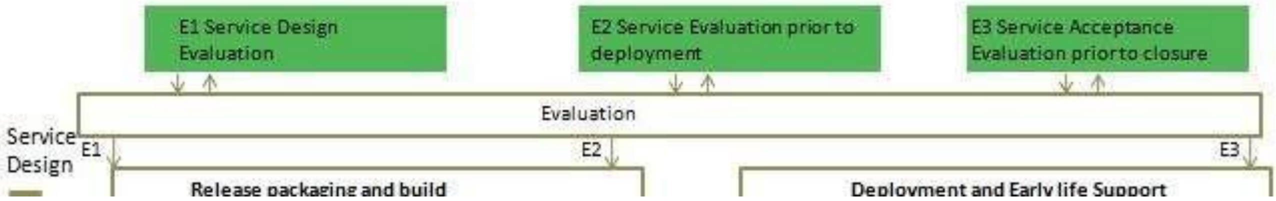
This process deals with closing a release after verifying if activity logs and CMS contents are up to date.

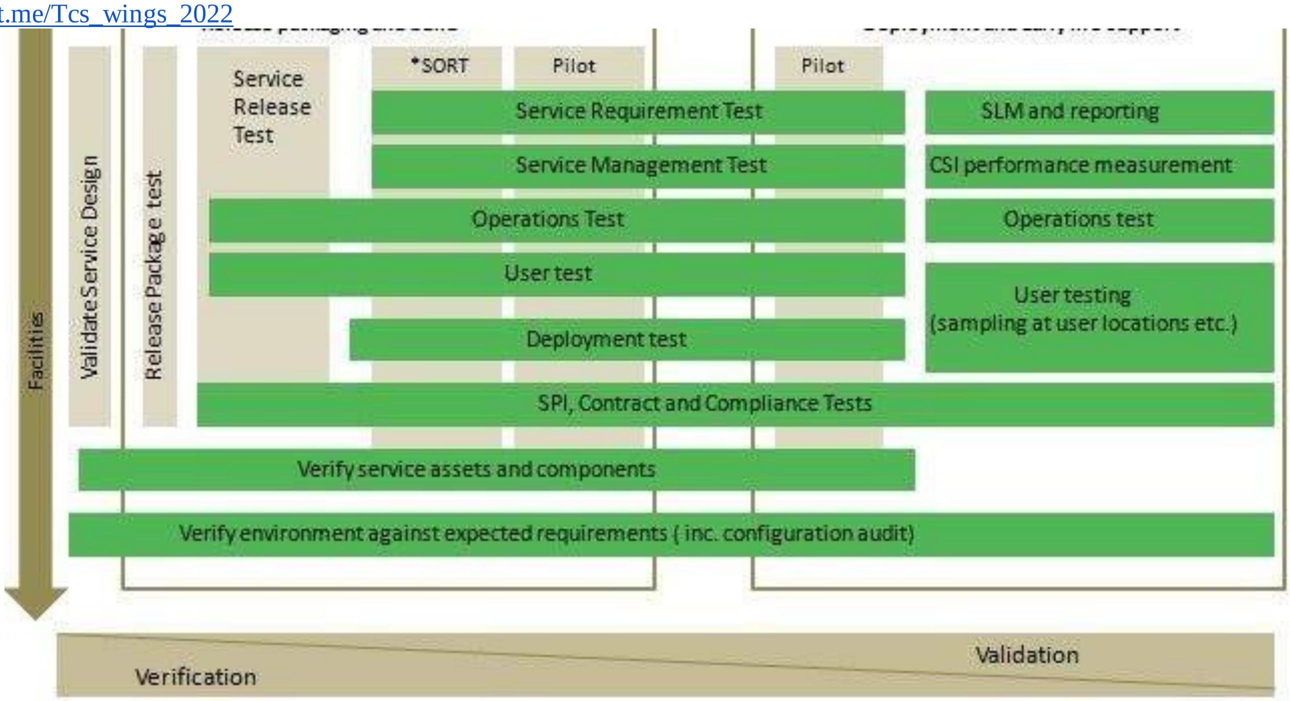
Validation and Testing

It is necessary to actively maintain test environments to ensure that the deployed releases meet the customer's expectations, and to verify that IT operations are able to support the new services.

> *Test Manager is the process owner of this process.*

The following diagram describes an example of service testing through the Service Transition stage of the lifecycle:


E1 Service Design Evaluation
E2 Service Evaluation prior to deployment
E3 Service Acceptance Evaluation prior to closure
Evaluation
Service Design
E1
E2
E3
Release packaging and build
Deployment and Early life Support



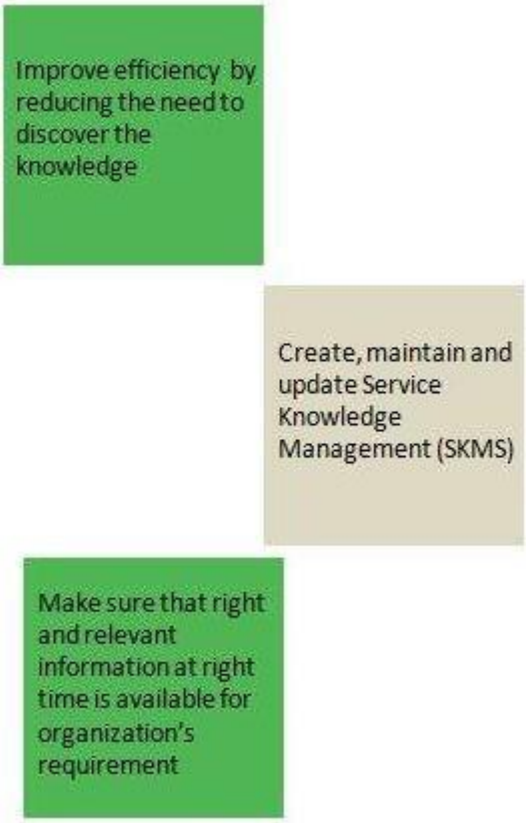
Knowledge Management

Knowledge Management deals with gathering, storing, analyzing and sharing the knowledge within an organization. It helps to improve efficiency by reducing the need to rediscover knowledge.

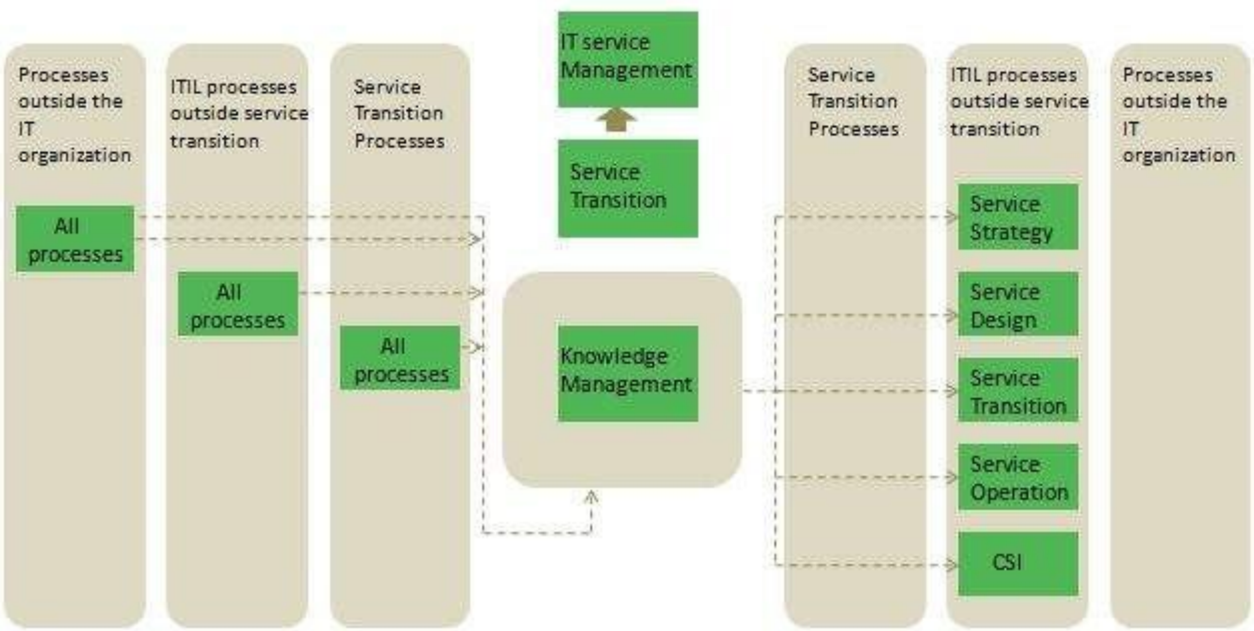
> *Knowledge Manager is the process owner of this process.*

Objectives

The following diagram describes the several objectives of knowledge management process:


Improve efficiency by reducing the need to discover the knowledge
Create, maintain and update Service Knowledge Management (SKMS)
Make sure that right and relevant information at right time is available for organization's requirement

Knowledge Management is a central process that receives input from all other processes and responsible for providing knowledge to service management processes as shown in the following diagram.


Processes outside the IT organization
All processes
ITIL processes outside service transition
All processes
Service Transition Processes
All processes
IT service Management
Service Transition
Knowledge Management
Service Transition Processes
ITIL processes outside service transition
Service Strategy
Service Design
Service Transition
Service Operation
CSI
Processes outside the IT organization

> *Service Knowledge Management System SKMS is a central repository of data, information, and knowledge that the IT organization needs to manage lifecycle of its services.*

Service Operation Overview

Service operationensures that services are being provided efficiently and effectively as per SLAs. It includes monitoring services, resolving incidents, fulfilling requests and carrying out operational tasks.

Key Points:

- Service Operation includes five process and four functions
- Service operation deals with day-to-day activities and infrastructure that are being used to deliver the services
- Service Operations is where all design and transition plans are executed and measured From
- customer point of view, Service Operation is where actual value is seen

Service Operation Processes

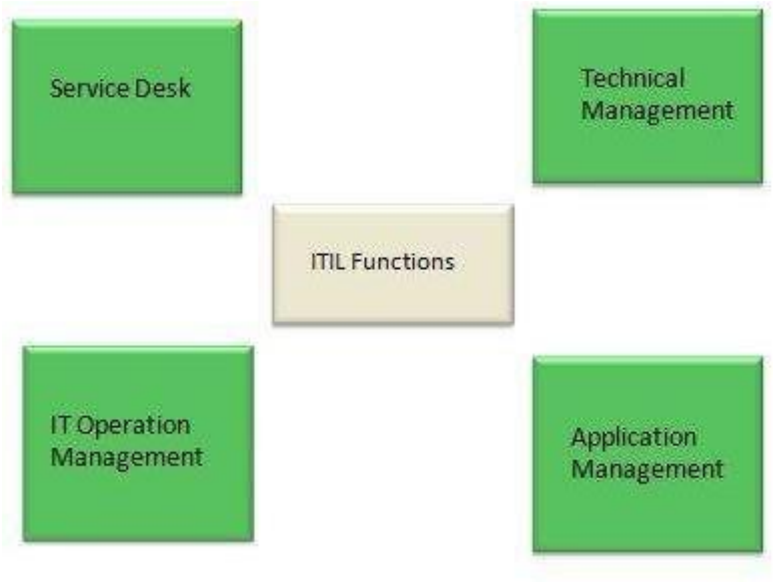
Service operations include total five processes as described in the following table:

S.N.	Process Description
1.	**Event Management** The objective of this process is to make sure all CIs are monitored constantly. It also filter and categorize the events in order to decide on appropriate actions.
2.	**Incident Management** The purpose of Incident Management is to restore the service to the previous stage as early as possible.

3.	**Request Fulfilment** This process deals with handling requests such as change password, create new user and create email id etc.
4.	**Access Management** This process deals with granting rights to authorized user to use the service.
5.	**Problem Management** This process deals with finding root cause of the problem and prevent incident to occur again.

Service Operation Functions

Service Operation comprises of four functions as shown in the following diagram:



Service Desk

Service Desk is the first and single point of contact. It plays vital role in customer satisfaction. It coordinates activities between end user and IT service provider team. It also own the logged requests and ensures closure of these requests.

There are four types of Service Desk:

Central Service Desk

In this, there is only one central Service Desk.

Local or distributed Service Desk

It is costly but closer to the user. It’s difficult to manage and maintain.

Virtual Service Desk

It is very costly and result in fast processing.

Specialized Service Desk

It includes dedicated skilled staff for specific queries.

IT Operation Management

This function is responsible for managing organization’s day-to-day operational activities

Technical Management

Technical Management staff includes technical expertises that are responsible for management of overall IT infrastructure.

Application Management

Application Management is responsible for managing applications and software throughout the lifecycle of the service.

Event Management

Event is defined as detectable occurrence that has significance for the delivery of IT service. Events are created by Configuration Item *CI* or the monitoring tools.

Event Management ensures that all CIs are constantly monitored and define a process to categorize these events so that appropriate action can be taken if required.

> *IT Operation Manager is the process owner of this process.*

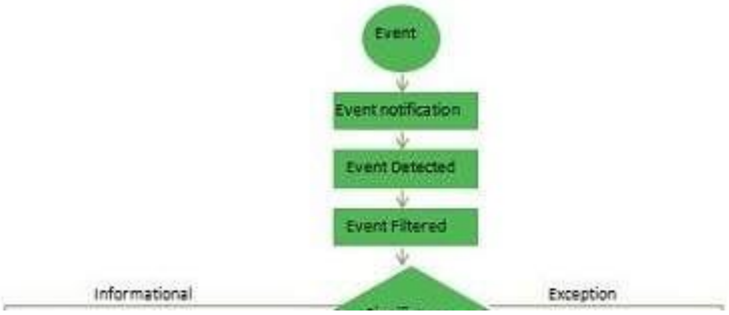
Event Management can be applied on the following aspects:

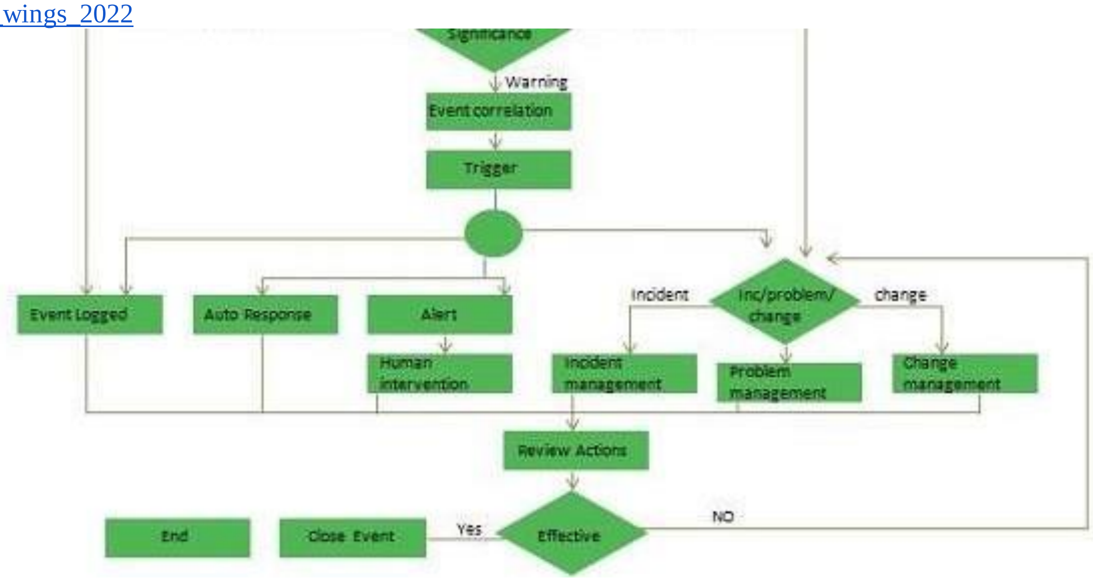
- Configuration Items *CIs*
- Security
- Environment Conditions *e. g. fireandsmokedetections*
- Normal activity *e. g. trackingtheuseofanapplicationorperformanceofaserver*
- Software licence monitoring for usage to ensure legal licence utilization and allocation

> *There are two types of monitoring tools as described below:*
>
> - *Active monitoring tools monitor CIs for their status and availability. Any deviation from normal operation is communicated the appropriate team for action*
> - *Passive monitoring tools detect and correlate operational alerts or communications generated by CIs.*

Event Management Process

The Event Management Process is self descriptive as shown in the following diagram:





Benefits

Here are the several benefits of Event Management:

- It helps to detect incidents at early stage. Thus incident can be assigned to appropriate team before any service outage occurs
- It removes need for expensive and resource intensive monitoring
- It can also be applied on some service management processes such as Availability management or capacity management. It automatically generates signal when status changes that allow appropriate team to perform early response
- Since it is automates, hence provides increased efficiency

Incidents and Requests

Incident is defined as any disruption in IT service. Incident can be reported either through the Service Desk or through an interface from event management to incident management tools.

Incident Management deals with handling incident and ensures to restore IT service soon as possible.

> *Incident Manager is the process owner of this process.*

Incident Models

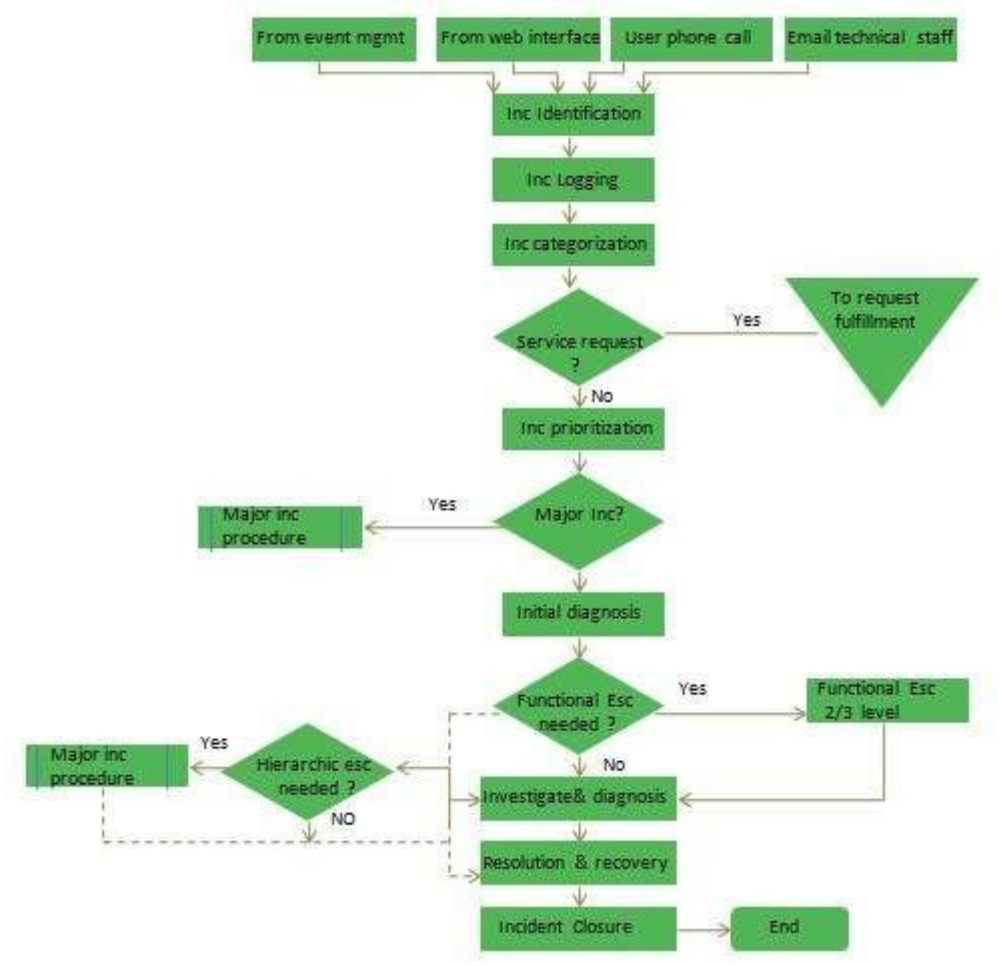
There are always some incidents which are not new. They may happen again over a period of time. Therefore it is best practice to have pre-defined model to handle such incidents.

An incident models should include:

- Steps that should be taken to handle the incident
- Chronological order these steps should be taken in, with any dependences or co-processing defined
- Responsibilities: who should do what
- Timescales and thresholds for completion of the actions
- Escalation procedures; who should be contacted and when Any
- necessary evidence-preservation activities

Incident Management Process

Following diagram shows several standard steps to be taken when an incident occurs:



Incident Logging

All incidents should be fully logged and date/time stamped.

Incident Categorization

It is important later when looking at incident types/frequencies to establish trends for use in Problem Management.

Incident Prioritization

It deals with severity of an incident: Low, Medium or High.

Incident Diagnosis

Service Desk analyst must carry out initial diagnosis while the user is on call and try to discover the full symptoms of the incident and to determine exactly what has gone wrong and how to correct it.

Incident Escalation

Function escalation

When it becomes clear that Service Desk is unable to resolve the incident or target time for Service Desk has been exceeded, the incident must be escalated immediately for further support.

Hierarchic escalation

Hierarchic escalation is done when either incident is serious in nature or 'Investigation and Diagnosis' is taking too long time.

Investigation and diagnosis

It includes the following activities:

- Understanding what exactly has gone wrong
- Understanding chronological order of the events
- Confirming the full impact of the incident
- Identifying any events that could have triggered the incident Searching
- for previous similar kind of incidents

Resolution and recovery

A potential resolution need to be identified, applied and tested.

Incident closure

Before closing an incident, Service desk should ask the user whether he is satisfied and agree to close the incident.

Request Fulfilment

Service Request refers to demand by the users. These requests can be regarding small changes, changing the password, installing additional software application, requesting information etc.

> *Incident is unplanned event but Service Request can be planned.*

Key Points:

- Depending upon number of Service Requests, an organization usually has, a specialized teams can be formed to fulfil those requests.
- For frequently recurring requests, a predefined model can be devised to fulfil the requests.

Problem Management

In ITIL, **Problem** is defined as unknown cause of one or more incident.

Problem Management ensures identification of problems and performs Root Cause Analysis. It also ensures that recurring incidents are minimized and problems can be prevented.

> *Problem Manager is the process owner of this process.*

Key Points:

- Problem Management comprises of activities required to diagnose the root cause of the incident and to determine the resolution to those problems
- When a problem is resolved after root cause analysis, it becomes known error

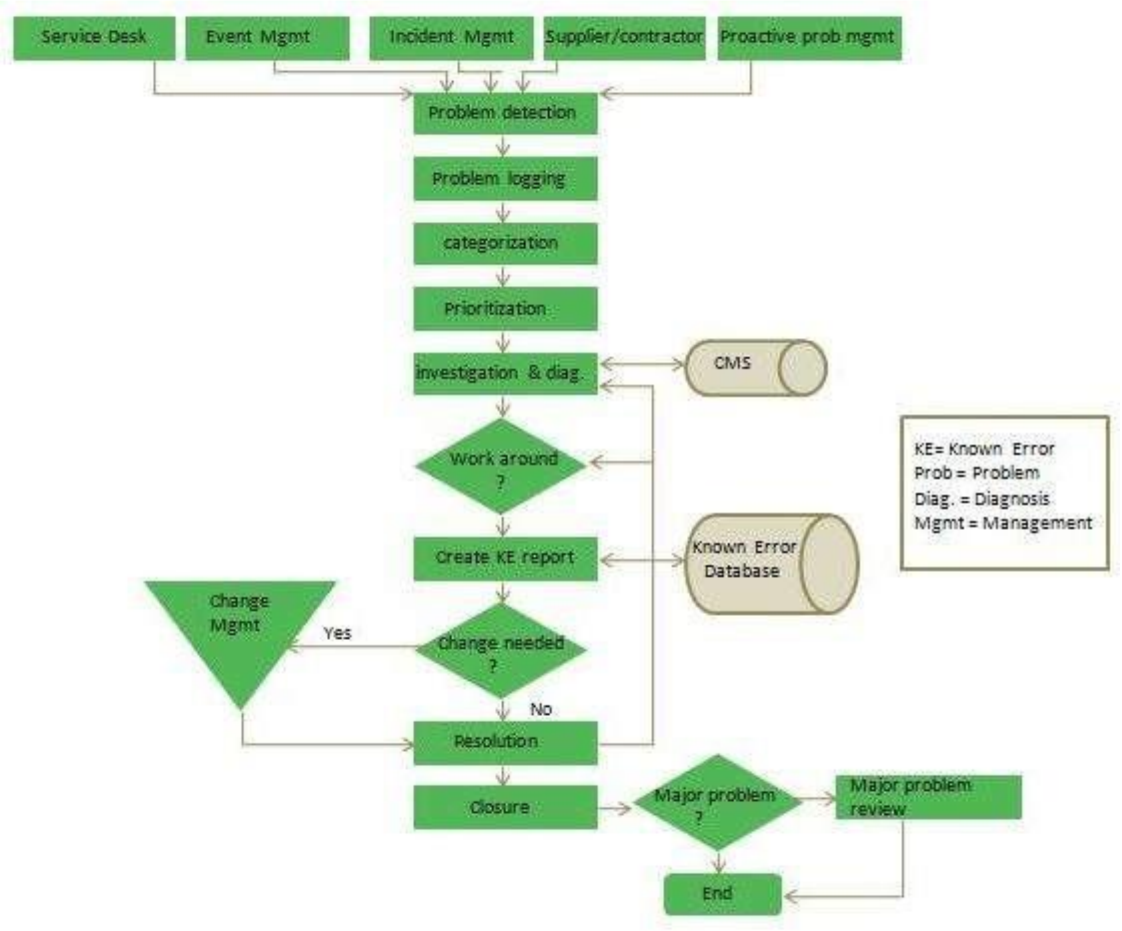
- Problem Management also records information regarding problems in a system called **Known Error Database *KED***

Problem Management consists of following two processes:

- Reactive Problem Management is executed as part of service operation
- Proactive Problem Management initiated in service operation but generally driven as part of Continual Service Improvement

Problem Management Process

The following diagram describes activities involved in Problem Management:



Problem detection

Problem can be detected in following ways:

- Analysis of incident by technical support group
- Automated detection of an infrastructure or application fault, using alert tools automatically to raise an incident which may reveal the need for problem management
- A notification from supplier that a problem exists that has to be resolved

Problem logging

Problem should be fully logged and contains the following details: User

- details
- Service details

- Equipment details
- Priority and categorization details
- Date/time initially logged

Problem categorization

In order to trace true nature of Problem, It is must to categorize the Problems in same way as Incidents.

Problem Prioritization

Problems must be categorized in the same way as incidents to identify how serious the Problem is from an infrastructure perspective.

Workarounds

It is temporary way to overcome the difficulties. Details of workaround should always be documented within Problem record.

Raising a Known Error Record

Known error must be raised and placed in Known Error Database for future reference.

Problem Resolution

Once resolution is found, it must be applied and documented with the problem details.

Problem closure

At time of closure, a check should be performed to ensure that record contains full historical descriptions of all events.

Major Problem Review

A review of following things should be made:

- Those things that were done correctly
- Those things that were done wrong What
- could be done better in future How to
- prevent recurrence

Access Management

Access Management deals with granting access to authorized access while preventing access to non-authorized users.

> *Access Manager is the process owner of this process.*

Key Points:

- Access Management is also known as ‘rights management’ or ‘identity management’

- Access Management process is executed by technical and application management functions.
- Access Management can be initiated by Service Request through Service Desk

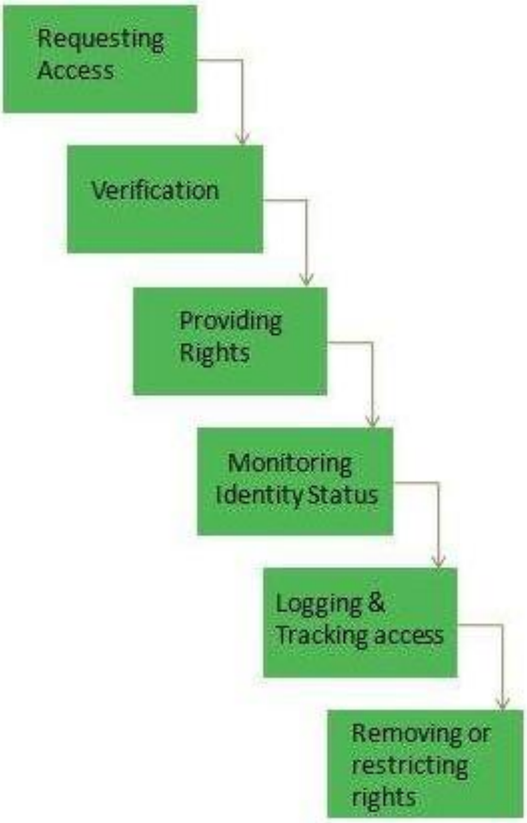
Value to business

Access Management adds value to business in following ways:

- Employees have right level of access to execute their jobs effectively The ability
- to audit use of services and to trace the abuse of services
- Controlled access to services ensures that organization is able to maintain more effectively the confidentiality of its information

Access Management Process

Activates involved in Access Management are self explanatory as shown in the following diagram:


Requesting Access
Verification
Providing Rights
Monitoring Identity Status
Logging & Tracking access
Removing or restricting rights

Continual Service Improvement

Continuous Service Improvement *CSI* deals with measures to be taken to improve the quality of services by learning from past successes and failures. Its purpose is to align and realign IT Services to the changing needs by identifying and implementing improvements to the changing business needs.

Objectives

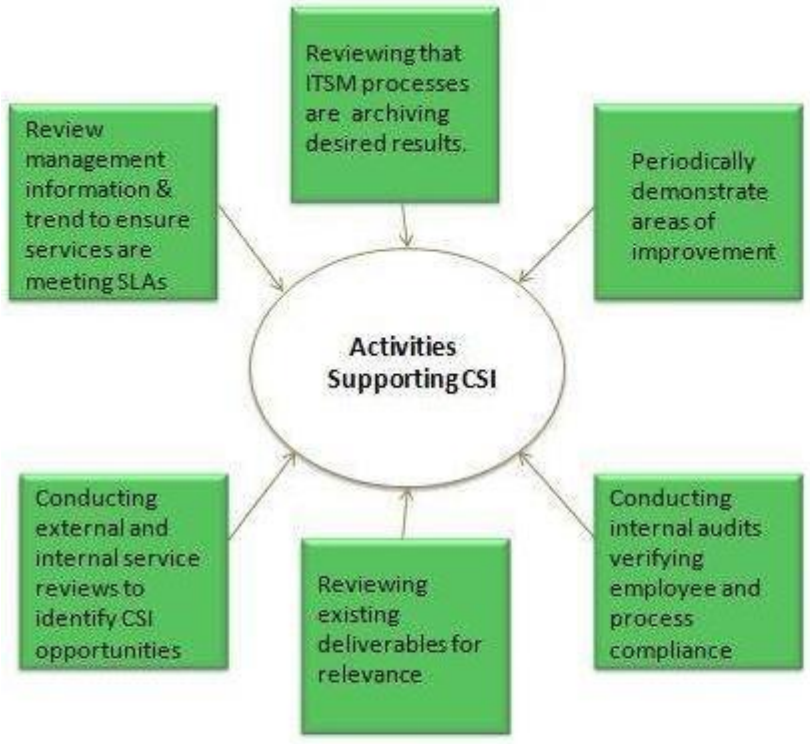
Here are the serveral objectives of Continual Service Improvement:

- Review and analyze improvement opportunities in each lifecycle phase Review
- and analyze Service Level achievement results
- Improve cost effectiveness of delivering IT services without sacrificing customer satisfaction

- Identify and implement individual activities to improve IT service quality
- Ensures applicable quality management methods are used to support continual improvement activities

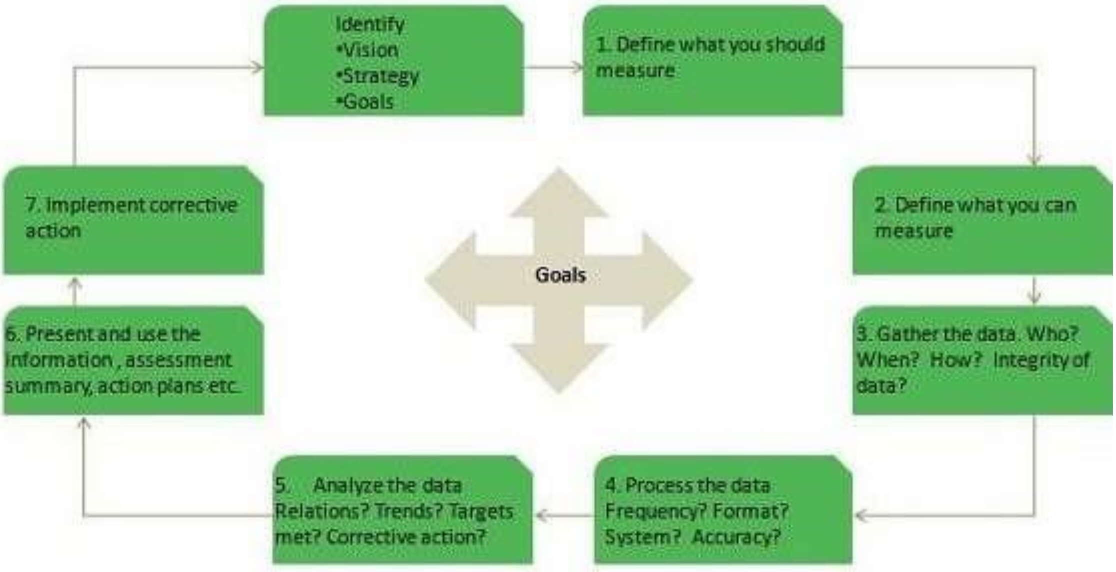
Activities Supporting CSI

The activities that support CSI do not happen automatically and thus required to be owned within IT organization which is capable of handling the responsibilities and processes the appropriate authority to make things happen. Here are the activities supporting Continual Service Improvement:


Reviewing that ITSM processes are archiving desired results.
Review management information & trend to ensure services are meeting SLAs
Periodically demonstrate areas of improvement
Activities Supporting CSI
Conducting external and internal service reviews to identify CSI opportunities
Reviewing existing deliverables for relevance
Conducting internal audits verifying employee and process compliance

CSI Process Model

The steps involved in CSI are self explanatory as described in the following diagram:


Identify
•Vision
•Strategy
•Goals
1. Define what you should measure
2. Define what you can measure
3. Gather the data. Who? When? How? Integrity of data?
4. Process the data Frequency? Format? System? Accuracy?
5. Analyze the data Relations? Trends? Targets met? Corrective action?
6. Present and use the information , assessment summary, action plans etc.
7. Implement corrective action
Goals

Perspectives of Benefits

Here are the four perspectives of benefits that are taken into consideration while discussing outcomes of CSI:

Improvement

This is comparison of result of current state of service to previous.

Benefits

It is the gain achieved through realization of improvements.

Return on Investment

It is the difference between benefits and amount expanded to achieve that benefit.

Value on Investment

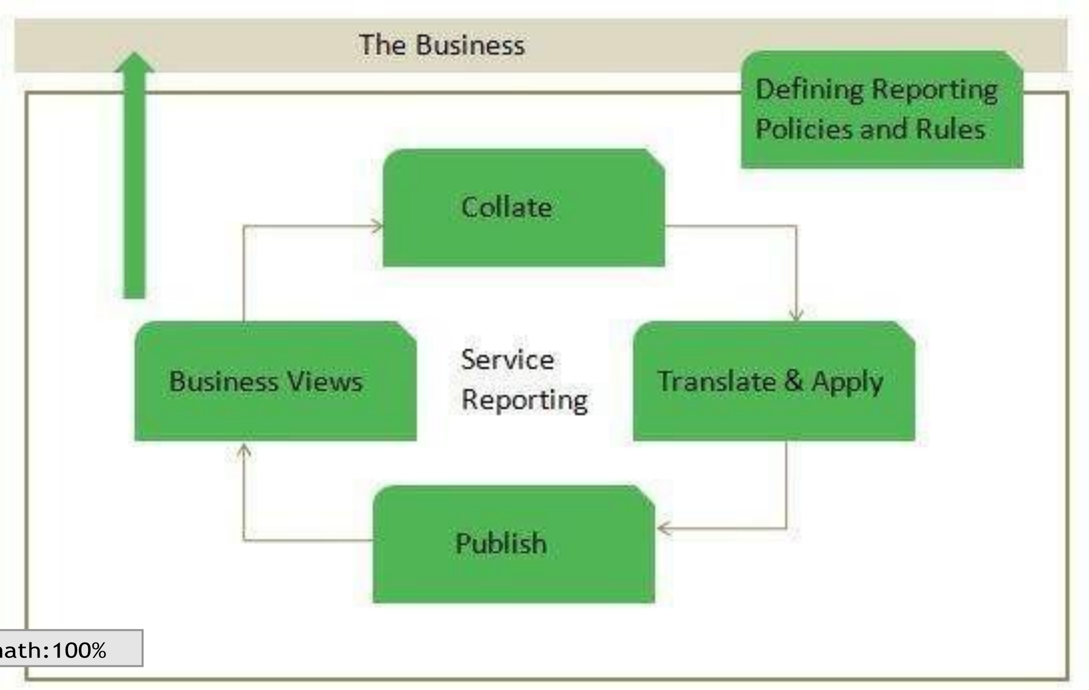
It is the extra value created by establishment of benefits that include non-monetary or long-term outcomes.

Service Reporting

Service Reporting deals with producing and delivering reports of achievements and trends against Service Levels.

> *It is best practice to generate reports as per agreed format, content and frequency with the customers.*

The following diagram describes the overall flow of Service Reporting process:



Processingmath:100%