Ojasalo Hannu

Release Management as the Backbone of Service Delivery for Business Critical Enterprise Applications

29.11.2013

1.0

TURKU UNIVERSITY OF APPLIED SCIENCES — ABSTRACT

Degree Programme: Master of Engineering	
Author(s): Hannu Ojasalo	
Title: Release Management as the Backbone of Service Delivery for Business Critical Enterprise Applications	
Specialization line: Technology Competence Management	Instructor(s): Olli Mertanen, Jaakko Hissa
Date: 29.11.2013	Total number of pages: 104

Application management services are initiated to take care of enterprise applications in the use phase of their lifecycle. Most important of the application management processes are incident, problem, change, knowledge and release management. The aim of this thesis of mine is to validate the position of release management.

Release management can be the backbone for enterprise application management. It makes sure that the risk of negatively affecting the operation of the application is mitigated to the full.

The release management process introduced in this thesis is developed to be light and adaptable. It can be taken into use with minimum effort. Despite this it can provide maximum benefits of visibility, predictability, risk mitigation efficiency and even financial benefits.

The introduced release management process enables the work to be outsourced. The main benefit for outsourcing release management is that the process is controlled by a dedicated organization, which does not take the control away. The risk can be divided with the service provider, which also gives the provider more motivation to make the process really work.

Keywords: Application management, release management, ITIL, ASL, process modeling, process development, enterprise IT, information systems, IT risk management, continuous improvement

Deposited at: Library, Turku University of Applied Sciences

TURUN AMMATTIKORKEAKOULU	TIIVISTELMÄ

Koulutusohjelma: Insinööri (YAMK)	
Tekijä(t): Hannu Ojasalo	
Työn nimi: Release management liiketoimintakriittisten tietojärjestelmien sovellushallinnan tukirankana	
Suuntautumisvaihtoehto: Teknologiaosaamisen johtaminen	Ohjaaja(t): Olli Mertanen, Jaakko Hissa
Opinnäytetyön valmistumisajankohta: 29.11.2013	Sivumäärä: 104

Sovellushallintapalveluita perustetaan huolehtimaan yritysten sovelluksista niiden elinkaaren käyttövaiheessa. Sovellushallinnan prosesseista tärkeimmät ovat incident, problem, change, knowledge ja release management. Tämän opinnäytetyön tarkoituksena on osoittaa release management –prosessin tärkeys.

Release management voi toimia koko yrityksen sovellushallinnan tukirankana. Sen avulla voidaan varmistaa, että riskit negatiivisista vaikutuksista sovellusten toimintaan voidaan hallitusti välttää.

Tässä opinnäytetyössä esitelty release management –prosessi on suunniteltu kevyeksi ja sopeutettavaksi. Tästä huolimatta sen avulla voidaan saada aikaan hyötyjä kuten muun muassa ennustettavuus, korkeampi riskienhallinnan tehokkuus ja jopa parempi kustannustehokkuus.

Esitelty release management –prosessi mahdollistaa sen release management -työn ulkoistamisen. Ulkoistamisen suurin hyöty on se, että prosessia hallinnoi siihen omistautunut organisaatio, joka ei kuitenkaan ota täyttä kontrollia prosessista.

Hakusanat: Sovellushallinta, release management, ITIL, ASL, prosessien mallinnus, prosessien kehittäminen, yritysten IT, tietojärjestelmät, infomaatioteknlogian riskienhallinta, jatkuva parantaminen

Säilytyspaikka: Turun ammattikorkeakoulun kirjasto

Contents

Pictures

1. Introduction

Enterprise applications have a lifecycle. When the applications are well and fully thought of, their lifecycle starts from creating a business case and ends to a graceful ramp down. Like living things, they require nurturing to thrive. We cannot just create an application and take it into use without any precaution and then just hope that it stays usable. When the business using the application is larger, the role of management of the application grows.

In use, an enterprise application has to be first and foremost available. Then it has to also have the required performance. But these things are nothing if there is no business case behind the application. It has to be a part of the daily operations or it is only a hindrance, which drags processes down. This is becoming more valid every day as the existence of every application is weighted in comparison to the costs they accumulate. IT is not the core business of many companies reliant of IT so less and less resources should be used maintaining it.

Application management services are initiated to take care of the applications in the use phase of their lifecycle. As with the other phases, it has to be planned. To help us with this planning work we have to rely mostly on our experiences from delivering such services before in the same environment or similar environments. But there are no right answers or solutions, we always have to adapt to meet the need at hand.

We can also utilize and adapt so called best practices. There are several such libraries available, which draw their information from a larger scope and base of experiences. From these the most notorious and utilized is the IT Information Library, ITIL. This set of best practices will be referenced in this thesis constantly among other relevant theories related to the topic.

Business critical applications add complexity to the already problematic area of application management. The above mentioned availability has to be priority one and performance is not far behind in importance. Good examples of business critical applications are logistics systems or factory floor applications. In theory in some cases bad usability of these business critical applications can cause significant financial losses. This brings the applications to the attention of the business. The IT department is not alone and has to demonstrate the capability to manage these applications in a planned manner.

There are many central processes in application management, which are also listed in ITIL. Most important of these are incident, problem, change, knowledge and last but not least release management. The last one I tend to think as the backbone, which ties all of these processes together to ensure a holistic view to the use phase of an enterprise application.

Release management is often neglected or not given enough emphasis. When talking about business critical applications, it should be the essence of application management. But still incident and change management are given the center stage. This is of course natural as these processes are in the forefront when talking about the revenue gained from application management services. As the core duty of IT should be developing new solutions and thus accelerating the business, just keeping the current solutions up and running is not enough.

The aim of this thesis of mine is to validate this position of release management and give valid arguments to my claim above. I will draw the conclusions from my daily work, from interviews with my colleagues and also from the available documentation, both internal and universally available. This work will be useful to my organization as I will also define in this thesis a Light and Adaptable Release Management (LARM) process, which can be utilized for any service. It can be used for example where there is no such process readily available.

I am an experienced professional in the field of application management who believes that nothing is ever ready. In the IT world, users, processes, applications and the management has to evolve constantly. My daily work revolves around these issues, so finding new process solutions is justified.

2. The Nature of Enterprise Applications

Before I can proceed to discuss about enterprise application management, we have to first establish what enterprise applications are and what are the challenges managing them. And before this we have to take a look at enterprise organizations themselves and their relation to information technology. This relationship usually presents the most challenges to both the applications itself and their management. The enterprise application landscape is usually complex, which presents an elevated challenge for application management. At the end of this chapter I am going present the most common examples of enterprise applications, or more officially enterprise information systems.

There are some terms that will be used throughout my thesis, which are appropriate to introduce at this point:

- Enterprise – Large Corporation, which has a common strategy, uses the same systems and common information. An enterprise is a large-scale organization that is involved in, and must orchestrate, more than one independent business processes (IRMA, 2011).
- Corporate IT – The IT department of an enterprise, which is responsible for maintaining the above mentioned systems and information.
- Vendor – A company, which provides IT services for the enterprise through its corporate IT.
- Application, information system – A vital computerized tool for the enterprise for supporting or driving business processes. Usually provided by the corporate IT.

These terms will be detailed more in the upcoming chapters of my thesis.

2.1. Enterprise Information Systems

An enterprise application is a software system that integrates the business processes of organization(s) to improve their functioning. Integration of business processes plays an important role in this definition (IRMA, 2011).

The nature of enterprise application has changed. As earlier the applications supported the business, they are now the foundation of many businesses. In other words, many enterprises are very reliant on their applications. Many critical processes are run on an application or a set of integrated applications. If this kind of application is unavailable, it can have severe implications to the performance of the business. Enterprises house both critical and non-critical applications. Critical applications are those, which support daily operations. Good example of this kind of system is for example a cashier system.

Traditionally, enterprises are categorized in large (LE) and medium and small (SME) sized enterprises (IRMA, 2011). Organizations have different needs for information systems and data processing, resulting in different IT infrastructures. Information processing in organizations has been defined as including the gathering of data, the transformation of data into information, and the communication and storage of information in the organization (Egelhoff W.G., 1991).

Louis J. Taborda discusses in his publication “Enterprise Release Management: Agile Delivery of a Strategic Change Portfolio” (2011) about the new term Enterprise 2.0. This term goes to describe how the so called Web 2.0 tools are coming to the enterprise domain and how they are reshaping forever the ways of working. As Taborda (2011) states, this is evidence about the enterprises’ willingness to adopt new methodologies and tools. So, when considering adopting release management, social media should be considered to be used as a tool in some form.

Internal IT capability is one big factor in how the enterprise applications are run. The IT capability can be divided into IT Intensity and IT Integration (Fyrvik T., Uthaug E.S., 2005).

Different organizations are differently IT Intensive (Fyrvik T., Uthaug E.S., 2005). This means that some companies are relying on IT more, for example there might be a situation where there is not a single task in the company, which is operated without any computerized means. This requires a complex system landscape, which stores vast amount of information used in the operations of the company.

IT Integration level describes how well the organization's information system landscape is integrated together (Fyrvik T., Uthaug E.S., 2005). This is usually a big issue in larger enterprises: their different functions might have different or even separate instances of the same solution for running the same kind of operations. Corporate IT organizations have known this and have lately started to take more control of their internal IT. At the same time, there is a pressure towards IT to leave the control of IT matters to the hands of different business functions. The only way to hold on to effective centralized IT is to provide quality services and applications, which suit the needs of the various processes operated in the organization. ERP systems are one example of a solution for this.

Whilst the organizational structure and day-to-day running of large companies are significantly more complex than those of SMEs, they both have "a critical need for coordination and control of business activities, which, in turn, is related to the complexity of the organization" (Grinyer P.H., Al-BazzazS., Yasai-Ardekani M., 1986).

Organization size affects the requirements for enterprise information systems. The organization size is comparable to the estimated user base for an application. The size of the user base affects directly what kind of availability, quality and performance requirements the application has.

The nature of the business activities also affects these attributes of an application. Some businesses are run 24 hours a day and seven days a week. This requires high availability from some applications. Some applications have to have a certain level of performance to satisfy the requirements of a business process.

These attribute to the challenges IT has to face daily. Corporate IT has four main management challenges (Fyrvik T., Uthaug E.S., 2005):

- Time: The delivery time, required capacity and effort, realization
- Money: The funds available for the provision of the services
- Quality: Quality of the services provided and the way they are monitored
- Expectations: The expectations of customers and suppliers and the agreements made with them

Time and more precisely timeliness is important. The Corporate IT has to provide solutions in a predefined time. These can be agreed by the business and IT before any solutions or services are delivered. Timeliness is a very important factor for business critical applications. They have to work as designed most of the time, which usually is 100% of the time. Time is also required as a resource to design and build the solutions.

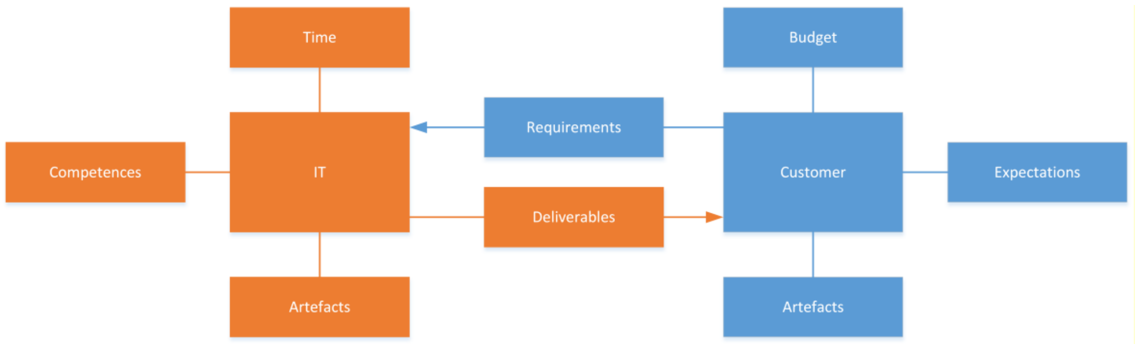
Everything is about money. Without a budget there can be no applications. They cannot be developed or maintained. No people can be allocated to support the applications. Usually there is no budget if the business is not performing as expected. When business is working fine and the financial performance is good, the applications are usually functioning as should. Then there is also budget to enhance the applications and develop new solutions. Which should be the main purpose of any IT department. Shamefully many IT departments use most of their time to fix faults in the existing applications. This is only beneficial to support the current use.

”Quality is king”. That is a catchphrase used often when there is money available. Unfortunately quality is the first thing, which is sacrificed when there is no budget. Solutions are delivered with haste and with low cost resources. Quality management functions are streamlined to minimum. The implications of this usually surface sooner or later in the business processes. It is an old truth that when you design and build with care and time, there are lesser problems.

Expectation management is often neglected. It would be good to know what the end user base or the stakeholders of the application are expecting from the application. Are they expecting process improvement or better performance? Are they seeing the introduction of the application negatively and wait for it to introduce also more complexity to their work? These expectations should be measured beforehand and the design and implementation of the application should address these expectations.

One final key challenge for understanding and building enterprise applications is understanding diverse and volatile stakeholder requirements (IRMA, 2011). This is emphasized in the current world where IT has to provide for the business: if IT delivers applications that do not work as expected, the business will not use the application maybe at all. In this case, the business might even build their own solutions. IT has a supportive role.

The IT challenges have been illustrated in Picture 1. The customer can be internal or external, the situation is the same. As business provides the budget, it expects IT to deliver and satisfy the requirements.



Picture 1. IT issues originate from the resources, objects and relationships of the IT environment.

2.2. Application Landscape Complexity

One assumption is that the complexity of enterprise applications is due to the following factors (IRMA, 2011):

- Increasing size of IT systems and the organization itself;
- The interactions between different IT systems;
- The involvement of many different organizations in the constructions and use of these IT systems; and,
- The increasing rate of organizational and social change.

Enterprises integrate organizations, departments, and even entire businesses to achieve shared goals. Processes within enterprises can benefit from IT infrastructure (IRMA, 2011).

As mentioned earlier, an enterprise can house many different applications, even for the same purpose. The corporate IT might not even be aware of the different solutions that are in use in the organization. This makes the application landscape to be fragmented and because of this very hard to coordinate. Corporate IT might have the responsibility of the enterprise wide IT, but the business is seldom bound to those IT rules. Those days are over when IT presented a solution and business had to use it. More and more business is defining the need and IT provides only the solution to satisfy it. IT is seldom the core business of an organization. Development is done process first, albeit IT has the ability to enhance the processes both performance and quality wise.

A high number of the applications in an enterprise application landscape can be integrated. They can be integrated to other systems in the landscape or the integrations can be to external systems. The integrations can be triggered by actions in the systems themselves or they can be timed to be run at a certain time. This depends on the type of the data. The integrated applications present a complex network, which can be very fragile. If no thought has been given to the whole landscape, this can present a big risk to the whole business.

A failure in a non-critical application can cause failure in a critical application if the landscape is not planned properly. The applications might be for example run on the same hardware environment. When one of these applications causes a resource overflow the whole landscape can be affected. For example network connections are very critical. Even one single user can cause significant harm if there are loopholes in the system.

The conclusion of the above is that there is really a need for the role of enterprise IT architect. Depending on the size of the organization, there might be even an enterprise architecture office in the organization. In this role they control the IT landscape and can give their suggestions, and in some case, approval to any changes implemented in the information systems. Release management could be integrated to the enterprise architect function.

A corporate wide release management strategy and policy would be beneficial to enterprises. Then the risk of critical failures and loss of revenue can be mitigated. This strategy would present the rules how different types of changes could be introduced, what is the approval flow and what standards should be followed. The policy would contain the rules how decisions are made. They would also act as a guideline for the individual release planning activities.

The strategy and policy could for example define what kind of team is required for different types of releases. The team for a maintenance release usually looks different from a quarterly release team. Depending on the organization, one team could be the sole responsible for all releases or there could be one release team for one application.

2.3. Examples of Enterprise Applications

When writing about enterprise applications, the first example is almost always SAP. It is the most widespread Enterprise Resource Planning (ERP) solution in the world (Columbus L., 2013). It is used by various different industries and has the capability to be the sole solution in an organization. It can support finance, logistics, sales, project and production processes among others.

The architecture of SAP is complex and requires a versatile field of expertise to support and develop it. One expert is usually proficient on one to three different areas of the application. This setting is the weakest spot of SAP and presents numerous challenges for application management. There are of course other ERP system vendors. For example Microsoft has seen some moderate success in the field lately.

Other enterprise applications include various product lifecycle management (PLM) systems, customer relationship management (CRM) applications, factory floor applications and for example the cashier systems mentioned earlier. These examples provide one more evidence for the fact that enterprise application management is challenging.

Another example in this area is the infrastructure being developed to support the National Health Service (NHS) in the United Kingdom where the information systems being developed to support management of patients' records and prescriptions can be considered as an enterprise information system (EIS), because such IT infrastructure aims to connect independent departments within and outside of the NHS. While we are looking at the NHS, which is a public sector organization, we can raise e-Government as another example of public sector organization that may be supported by and hence benefit from EIS infrastructure because it connects various governmental organizations or departments together to let information flow seamlessly between them. (IRMA, 2011)

One very interesting current topic in enterprise information system area is cloud based applications. This kind of applications reduce many risks related to information systems but the down side is that they elevate the information security risks. There are many popular cloud based enterprise applications, which can also be critical to the business. Some enterprises might rely heavily on a CRM system from which Salesforce.com is a good example.

3. Application Management as a Practice

In this chapter I will proceed to take an overview to application management (from here onwards referenced also as AM). This requires a brief look to the history of the doctrine and what it is today. AM is practically nothing without ITIL and other related best practice libraries, so they are discussed here too. The main AM processes – incident, change, problem, knowledge and release management – are introduced. The latter process is discussed in more detail in the following chapters.

3.1. What is Application Management?

By the ITIL definition, application management (OGC, 2007)

- is the custodian of technical knowledge and expertise related to managing applications. In this role application management, working together with technical management, ensures that the knowledge required to design, test, manage and improve IT services is identified, developed and refined.
- provides the actual resources to support the ITSM lifecycle. In this role, application management ensures that resources are effectively trained and deployed to design, build, transition, operate and improve the technology required to deliver and support IT services.

Application management has existed as a sector of IT about four decades now (ASL Foundation, 2004). Basically it has been all about managing, maintaining and enhancing information systems. Application management hasn't always been the term used to describe these activities, but suits them perfectly nonetheless.

The objectives of application management are to support the organization's business processes by helping to identify functional and manageability requirements for application software, and then to assist in the design and deployment of those applications and the ongoing support and improvement of those applications. (OGC, 2007)

The application lifecycle from creating the actual business case to the actual implementation into use is usually handled by a project or a program, depending on the size of the application. An application has always some justification to its existence. Usually the reason is that applications are created to support or manage some business process.

The road to a finished and deployed application is usually rocky: there are battles over requirements, investigations on bugs that seem to persist despite any effort to squash them, performing test cases that do not seem to pass and endless arguments and discussions where to go next.

Whatever the end result is, it is usually deployed to be used by a defined group of people. When the project releases application from its intensive care, it has to transition it to the continuous care in form of application management services.

AM services manage the application to the end of its lifecycle. One central task of application management is to handle any deviations in the system. These are usually called incidents or problems. Incidents are some deviation in the system that prohibits at least one user to follow a process. Problems are some underlying issues in the system that do not prohibit users to do their work but may cause incidents to occur.

AM also handles changes and enhancements. The application can and is usually enhanced to suit the ever changing needs of the enterprise business. These enhancements might be done by a separate project or application management can implement them. There are also changes that basically do not bring anything new to the user experience but might enhance the application in some other way. For example changes can be made to improve the performance of the application.

These enhancements and changes bring us to the topic of release management. It is used to manage the deployment of the changes so that the effect to the use of the application is minimal. That also underlines the ever defining role of IT: it has to exist but it should not be seen.

All of the central AM processes are linked or even tied together. The chain can start from incident management and lead to problem management. To fix some problem we need to initiate the change management process, which is in turn controlled by the release management process.

There are also a selection of supporting processes available, which are described in more detail in chapter 4.4: requirements, risk, test, configuration and knowledge management. These are discussed more in conjunction with release management as it benefits from these supporting processes the most.

The level of implementation of these processes depends on the services. In some services only incident and change management can be in use. Release management is quite often left out of the scope totally. But basically all of these are required for effective application management. They are actually present in all AM services although they might not be officially defined and used. For example the problem management process might be encapsulated in the incident management process. Then there is a risk though: solving the actual root cause might postpone the solving of the incident. That might lead to unwanted results.

3.2. Best AM Practices

The AM creed, as the whole IT creed, is very dependent on experience. There is no complete set of guidelines or written down theory how AM work should be approached. Every case has to be evaluated separately and the processes have to be applied accordingly.

But there is no need to start from scratch as there are several of IT related best practice libraries. As the name implies, these libraries have been gathered from the experiences of the professionals working in IT or application management. These libraries can be then referenced when building up services. It is though good to be aware that the content of these libraries cannot be and should not be applied as such (Myatt M., 2012).

Good examples of IT best practice libraries are COBIT, TOGAF BiSL, CMM, SFIA, ASL and ITIL. The two latter ones are tightly linked to application management. From these ITIL is universally known IT Service Management best practice library and it is also highly referred. In this chapter we will concentrate mainly on the best practices that ASL and ITIL provide for us to use in AM services.

Application Services Library, ASL, is an application management specific framework of a Dutch origin. It has been developed to give transparency to the AM services, enhance the management of costs, increase transferability and compatibility, increase flexibility and reliability and create uniformity of application management (ASL Foundation, 2004). The central concept for ASL is the service team, which acts as the single point of contact for the AM services.

IT Infrastructure Library, ITIL, is the most widely acknowledged and used IT related approach to IT Service Management. It is not only AM-specific as it can be applied to other types of IT services as well. ITIL is divided to five sections, which are service strategy, service design, service transition, service operation and continuous service improvement (OGC, 2007). The creation of ITIL was initiated by the government of the United Kingdom but it is currently privately maintained and developed.

ASL was created to fill in the gap that ITIL had regarding application management (ASL Foundation, 2004). In recent versions ITIL has added more emphasis to AM and thus ASL has been somewhat sidelined (OGC, 2007). ASL is still being developed by the ASL BiSL Foundation and is so kept relevant to the industry.

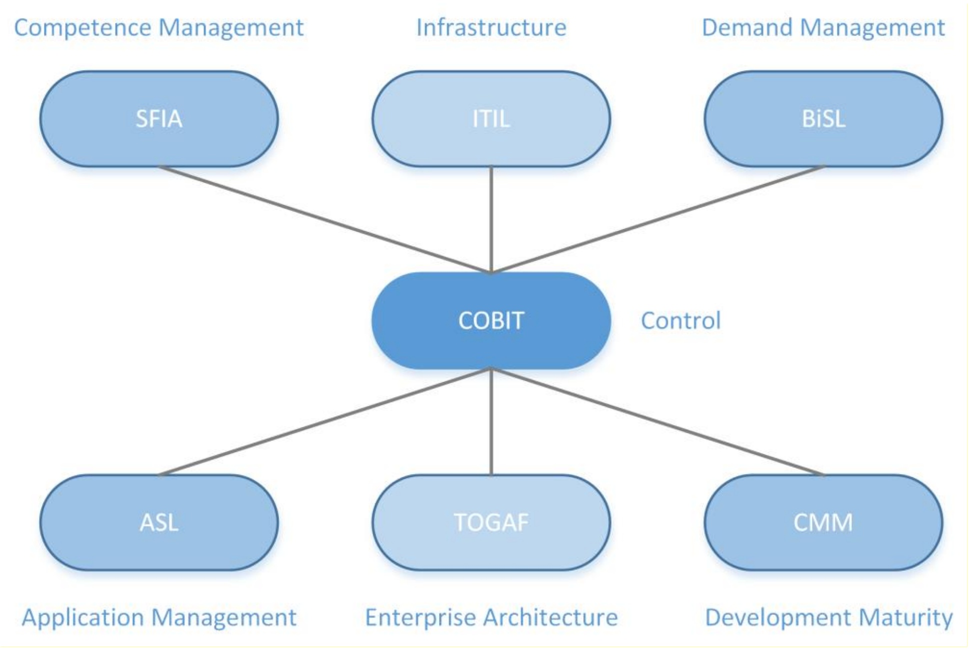
Both frameworks recognize added value in the other. The demarcation between customer (the business) and supplier of IT services is more explicitly drawn in

ASL than in ITIL. This gives a different perspective, which can be of added value. (Meijer M., Smalley M., Taylor S., 2008)

Both models address strategic aspects: ITIL addresses the generic service strategy while ASL focuses on the application strategy using process descriptions. The ITIL publications give sufficient guidance for organizations that manage commercial-off-the-shelf applications but if an organization maintains the applications and therefore actually modifies the source code, then ASL provides additional and necessary guidance. (Meijer M., Smalley M., Taylor S., 2008)

From the other above listed libraries (see picture 2)

- COBIT is about controls,
- TOGAF about enterprise architecture,
- BiSL is for demand management,
- CMM is for application development maturity definition and
- SFIA is for skills management.



Picture 2. COBIT has a central role in defining control to the IT processes.

These are all relevant frameworks when discussing release management. ITIL and ASL will be discussed below in more detail but the other will be referenced in relevant parts of this thesis as well.

There are many other frameworks available that are not named here and they are used for different areas of IT. There is business built around all of these, as trainings, literature, happenings and certifications are offered for them. For example some of the certifications are very valued and are extremely hard to achieve.

3.2.1. ITIL

The IT Infrastructure Library was established by one British government agency in the 1980's. As the practice of service management grew, so too did the dependency of the business. The aim was to develop an approach for efficient and cost-effective use of IT resources by British public sector organizations. The approach was to be independent of any supplier. ITIL grew from a collection of best practices to a framework observed in the IT service industry. (OGC, 2007)

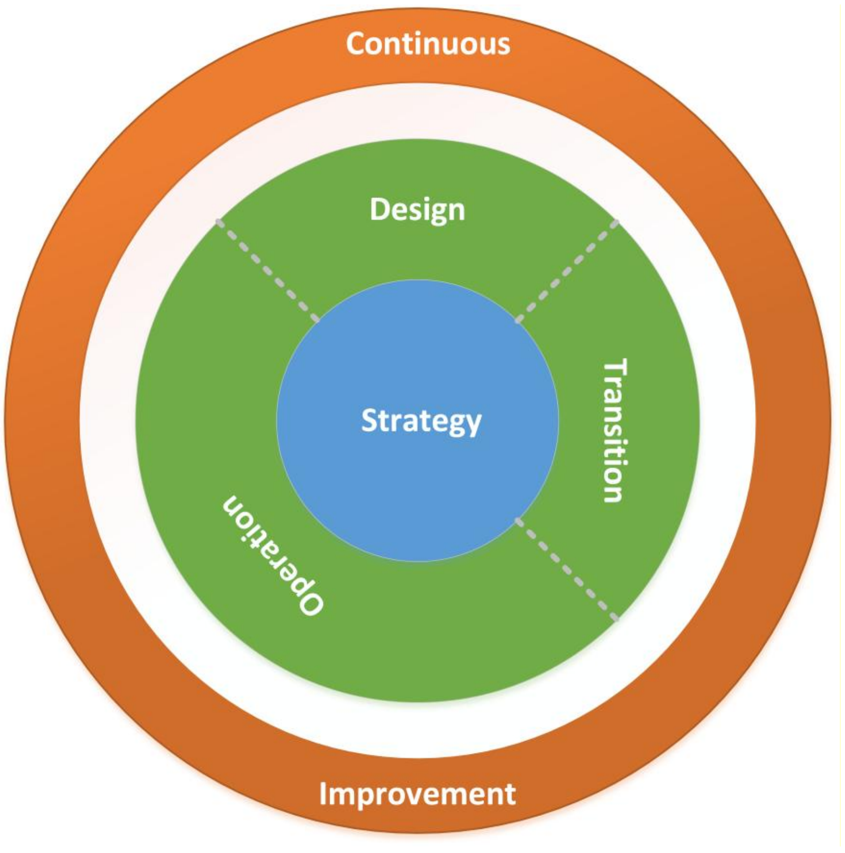
ITIL has been the basis of many frameworks developed by different commercial companies. This is one reason why it is currently so relevant and also so utilized. (OGC, 2007)

The current version of ITIL is 3, which has been available for years, but has been updated regularly. ITIL is a set of best practices for IT Service Management and it can be adapted to any IT service, including application management services.

The key characteristics of ITIL that contribute to its global success (OGC, 2007):

- It is non-proprietary: ITIL service management practices are applicable in any IT organization.
- It is non-prescriptive: ITIL practices have applicability to all types of service organizations.
- It delivers best practice: ITIL represents the learning experiences and thought leadership of the world's best in class service providers.
- It drives good practice: the practices ITIL represent are not all universally best practices but they are good practices.

ITIL is divided into five different processes (see picture 3), which follow the lifecycle of a service: service strategy, service design, service transition, service operation and continuous service improvement. The service strategy is the core. Service design, service transition and service operation are continuous processes, which are supported by the continuous service improvement process.



Picture 3. The core of ITIL is the Strategy process. In the picture the Operation process has been given more emphasis. Service Operation is where the customers see the most value

The service strategy defines the best practices for laying out the basics for the service. It answers the question of how would one go about deciding on a strategy to serve one's different customers? In a nutshell it provides best practices for assessing and developing the service strategy, defining the market space, defining service portfolios and catalogues, organizational positioning of the services, service outsourcing, service profitability and financial management of services. (OGC, 2007)

In the service design phase, the service or changes to the service are planned in detail. This phase provides alternatives for delivery models. One of the most important aspects of the service, service level agreement model, has to be defined in the service design phase. Customers have different requirements (Service Level Requirement, SLR) for service levels and they have to be accounted for every service. The way of monitoring the service level performance and other KPI setting has to be done in this phase. The availability plan and supplier management plan have to be created in the design phase. These are the most significant issues for application management in the design phase, it has additional content that is more applicable to other kinds of services. (OGC, 2007)

Service transition describes the best practices for the implementation of the service or changes to the service into the operation phase. Transition phase ensures that the service can be managed, operated and supported in accordance with the requirements and constraints specified within the service design phase. In transition phase it is crucial to plan and manage the resources required to provide mechanisms, package, build, test and deploy a release into production and establish the service specified in the customer and stakeholder requirements. (OGC, 2007)

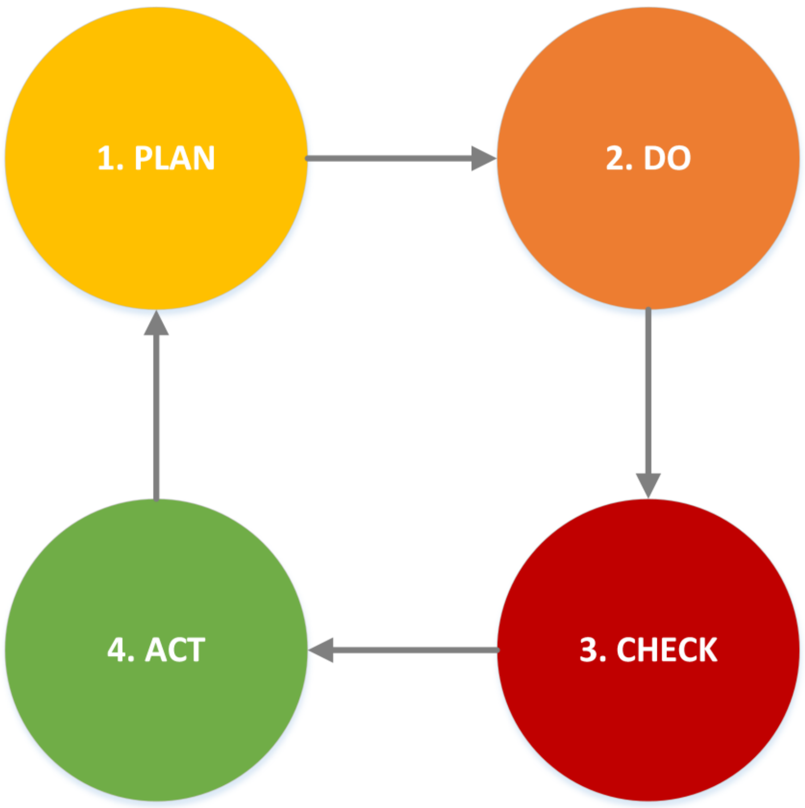
The transition process provides also a consistent and rigorous framework for evaluating the service capability and risk profile before a new or changed service is released or deployed. It also provides good-quality knowledge and information so that change and release management can expedite effective decisions about promoting a release through the test environment and into production. (OGC, 2007)

The service operation is the most important phase. It defines the best practices for the application management processes including incident management and the related processes. The purpose of service operation is to coordinate and carry out the activities and processes required to deliver and manage services at agreed levels to business users and customers (OGC, 2007). Service operation is also responsible for the ongoing management of the technology that is used to deliver and support services (OGC, 2007). From a customer point of view the service operation process is where the service value is seen. Service operation is dependent on detecting any deviation from normal or expected operation. The service operation processes are discussed in more detail in the chapter 3.3.

The continuous service improvement (see picture 4) phase defines how the services are measured and how the results can be used for improvement purposes. Its primary purpose is to continually align and realign services to the changing business needs by identifying and implementing improvements to IT services that support business processes. The ITIL service improvement process is divided into seven steps (OGC, 2007):

- Define what you should measure.
- Define what you can measure.
- Gather data. Who? How? When? Integrity of the Data?
- Process the data. Frequency? Format? System? Accuracy?
- Analyze the data. Relations? Trends? According to plan? Targets met? Corrective action?
- Present and the use the information, assessment summary, action plans, etc.
- Implement corrective action.

Between the cycles the vision, strategy, tactical goals and operational goals have to be identified and defined. Input to this can be service level requirements and targets, the service catalogue, organizational goals, legislative requirements, governance requirements, budget cycle and balanced scorecard. (OGC, 2007)



Picture 4. The improvement cycle, which is also known as the Deming Cycle.

Release management is defined in ITIL as one of the main processes in the service transition phase. ITIL defines release management's aim to be to build, test and deliver the capability to provide the services specified by service design and that will accomplish the stakeholders' requirements and deliver the intended objectives (OGC, 2007). Actually the process name is release and deployment management, but we are leaving the actual deployment part out of the discussions. ITIL describes those aspects of release management where service operation staff will be involved in service operation phase.

3.2.2. ASL

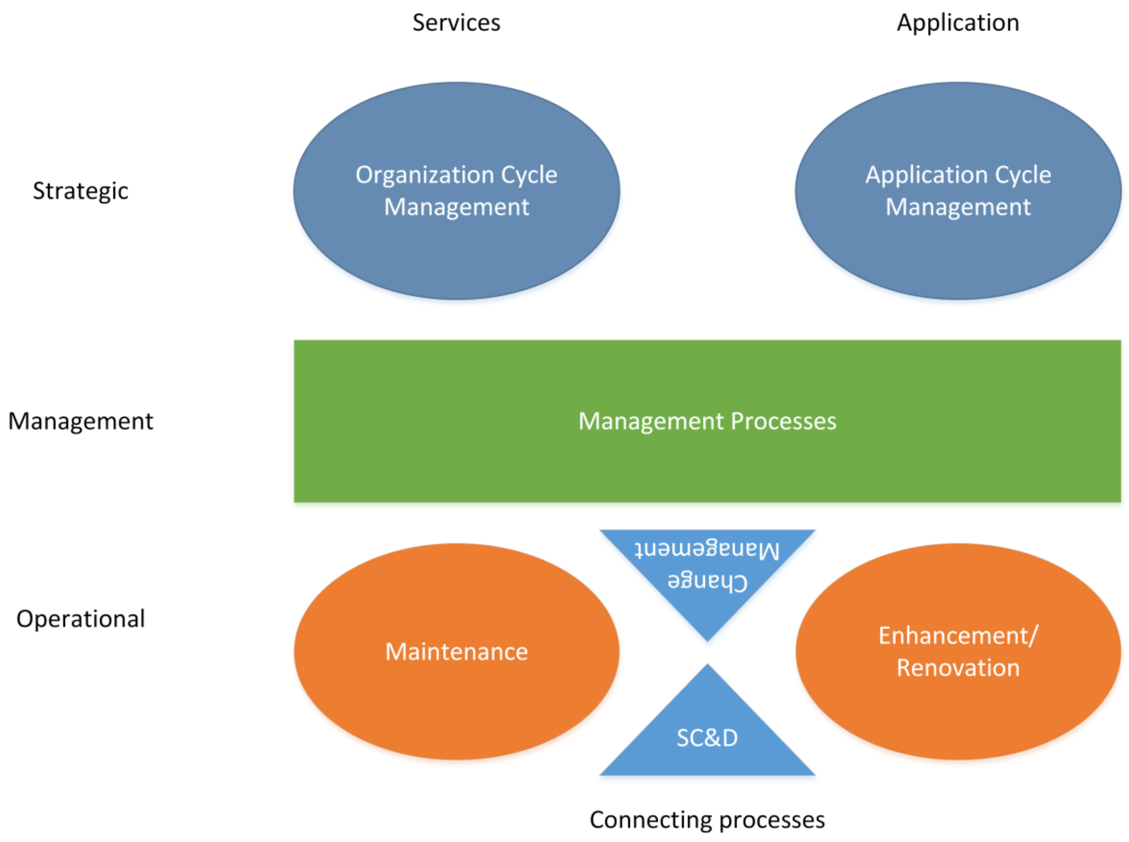
The Application Services Library was developed to fill in the gap that ITIL had for application management. Later also ITIL has been augmented with more application management best practices.

The ASL model is divided into strategic, management and operational levels (see picture 5). It is divided to six clusters of processes. The ASL framework uses two criteria to form these clusters (ASL Foundation, 2004):

1. The distinction between service processes and applications. The applications approach is distinct from other forms of management, such as technical management.
2. The distinction between service strategic (policy setting), management and operational processes.

The strategic level is divided into Applications Cycle Management (ACM) and Organizational Cycle Management (OCM). These sets of processes are used to define and validate the need and feasibility of the service (OCM) and the actual application (ACM). OCM is the cluster of processes, which define the future of the services and structure of the application management organization. ACM is concerned with the future supply of information of the lifecycle of objects (applications). This occurs at two levels: the individual application level and the application portfolio level.

The management processes are the center of the ASL framework. The management level houses the management processes: planning and control, cost management, quality management and service level management. These processes address the management issues application management can have. The operational level is divided into maintenance, enhancement and to the connecting processes. The maintenance part contains the processes required to run the current version of the application (incident, availability, continuity, capacity and configuration management). The objective of the maintenance processes is to ensure that the applications are operated and used in the best possible way to support the business processes, using a minimum of resources and causing the least disruption to the organization.



Picture 5. The ASL processes are divided into three layers: strategic, management and operational. (ASL Foundation, 2004).

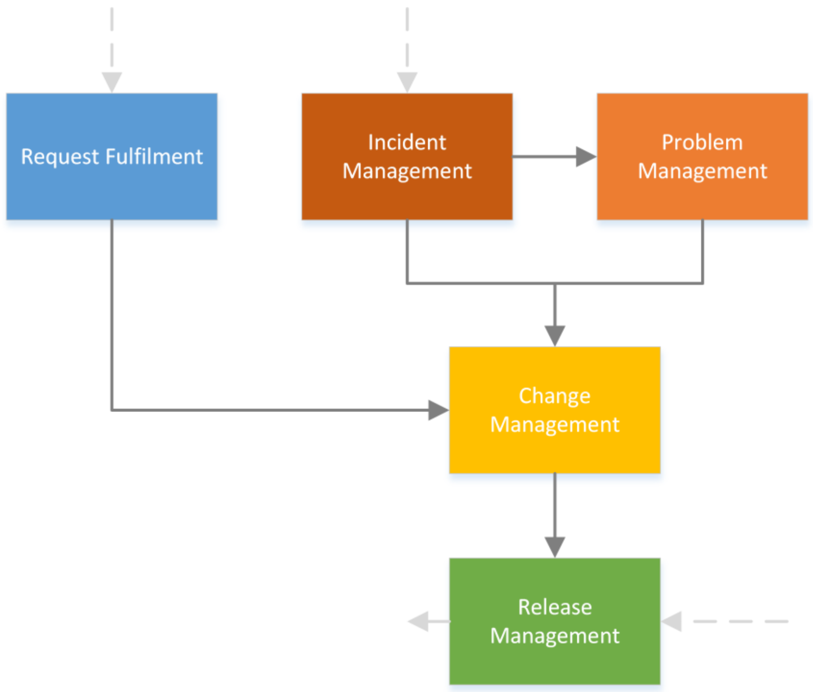
The enhancement processes are used for developing new functionalities to the application (impact analysis, design, realization, testing and implementation). Enhancements can involve a significant amount of work when major changes are made or if applications are continuously being improved: this will be done in form of projects. (ASL Foundation, 2004)

The connecting processes provide the synchronization and coordination between the two operational process clusters (ASL Foundation, 2004). They establish delivering the feedback from maintenance to be used as basis for planning enhancements. The connecting processes provide also a way to present the new enhancements as changes to the application, which are then maintained by the maintenance processes.

ASL omits the term release management totally. The equivalent in ASL is the software control and distribution process. It aims to provide the right application objects (or information about them) to the right processes, at the right time (ASL Foundation, 2004). The process has to provide secure procedures to limit the risks of unauthorized access, change or deletion (ASL Foundation, 2004). Part of the responsibilities of traditional release management are in the enhancement process.

3.3. The Main AM Processes

As listed earlier, the main AM processes are incident, problem, change and release management (see picture 6). They are all connected and form a chain from incident management to release management. They are not an isolated family of processes but are linked to other IT processes as well. There is also the process of request fulfilment, where so called service requests are worked on. It is an important source for enhancements delivered via release management.



Picture 6. The contents and triggers for release management originate in the other AM processes and other IT processes.

3.3.1. Incident Management

ITIL definition for an incident (OGC, 2007):

An unplanned interruption to an IT service or reduction in the quality of an IT service. Failure of a configuration item that has not yet impacted service is also an incident.

An incident is an anomaly in the use of an application. The incident can be detected by anyone, but usually it is related to a process, which cannot be completed because of the incident. Incident is usually created via the call management process. This process can include threads for different kind of input methods, like an email or a phone call. Call management makes sure that that all relevant information is available. This is usually the domain of a service desk function.

An incident can be resolved by applying a workaround to the process so that it can be completed. The actual root cause of the incident should be resolved in the problem management process. The root cause may require a change to be appled to the application. That change can be implemented into use via release management. This already demonstrates how the AM processes are linked together.

ITIL definition for incident management (OGC, 2007):

Incident management is the process for dealing with all incidents; this can include failures, questions or queries reported by the users, by technical staff or automatically detected and reported by event monitoring tools.

The most important details of an incident are its categorization, priority and the description. This has to be done to have an effective incident management in use. The most optimal situation is that the incident has all the required information submitted. This helps resolving the ticket in a correct and timely fashion. If information is missing, time is required to acquire it to successfully solve the incident. The less time there is for the resolving work the lower the quality of the solution is.

The incidents are usually prioritized. The priority of an incident is determined by its criticality and the impact it has on the use of the application. The prioritization tells the resolving party that in which order the incidents should be resolved. Universal prioritization classes are in the criticality order urgent, high, medium and low. Usually these different classes are given different SLA times.

The incident management is the most utilized and recognized of all of the AM processes. It is the first AM process to be implemented in the most cases. It is also the process that can be implemented in a wrong way. It can encompass both problem and change management. This makes the processes harder to monitor and manage. The slow separation of these processes is usually dependent on the unrecognized return of investment.

3.3.2. Request Fulfilment

The term service request is used as a generic description for many varying types of demands that are placed upon the IT department by the users. Many these can actually be small changes or may be just a question of requesting information. Service requests are handled separately from the incident and change management processes as they are not usually critical to the business.

Service requests are a lot like incidents, but they are not usually about any malfunction of the system. A service request could be initiated because there might be a need to fix some documentation about the application. Or for example a feasibility analysis could be requested via a service request. SLA is not usually applied to service requests.

3.3.3. Problem Management

ITIL defines problem management in its service operation process (OGC, 2007):

a problem is the unknown cause of one or more incidents.

The problem management process is usually interlinked to the incident management process or is mistakenly seen as part of it. Initiating the problem management process should lead to the fixing of the root cause of the incident. This fixing could be done either by a configuration action or a change to the application.

Problem management includes analyzing the problem and also comparing it to the known error database. If the problem is a known error, then the known solution is applied to it. If there is no known error corresponding to the problem, a new known error is created including the solution.

Problem management is not just for fixing the root causes of incidents. There should be continuous parallel process in which problems are sought out proactively. Proactive problem resolution is an effective way to remove any problems before they are realized as incidents.

3.3.4. Change Management

The enterprise applications are not static and they cannot be. They need to be constantly enhanced to meet the business needs. The Change Management process is used to manage changes to the applications.

There can sometimes be a distinction to change and enhancement management. The latter deals with changes that are not based on any incident or problem. The enhancement management process can be initiated separately. It can be used to deliver business requested changes to the application.

Because it focuses on details, change management is bound to a tactical time frame rather than a strategic one. Most of the changes being managed are scheduled a week or two out, with some exceptionally large or complex changes scheduled months in advance. Normally the change-management process dictates a minimum delay between proposing a change and executing that change, but change management also recognizes that urgent changes sometimes are necessary. (Klosterboer L., 2009)

Of all the service transition processes, change management is the most intimately related to service operations. Long before the days of ITIL and formally recognized best practices, every IT operations department had at least some form of change control in place. Every IT person has recognized that the best way to solve problems with permanent resolutions is through organized and controlled change to the environment. Unfortunately, every IT manager has also regretted making a change in which the details were not examined or controlled closely enough. (Klosterboer L., 2009)

3.3.5. Release Management

The aim of release management is to lower the risk of impact that changes could have on the application and therefore also to the business. Without proper release management, the introduction of new enhancements or totally new applications would affect the business processes and also other applications. The IT infrastructure has to be ready with the proper capability and also the introduced new or changed components have to be aligned with the IT infrastructure policies.

In reality, a release can contain both software and hardware components. Some new software functionalities may have new hardware requirements. These are called non-functional requirements. A good example situation is that when the hardware should still maintain the same level of performance despite of the grown amount of simultaneously run processes. More capacity should be added to the hardware. The hardware components should be installed or upgraded before or during the release deployment.

Despite this, release management is still software centric. But it has to be also noted that all of the software is not in the responsibility area of application management. For instance operating systems, database software and core application components are often not maintained by an application management function.

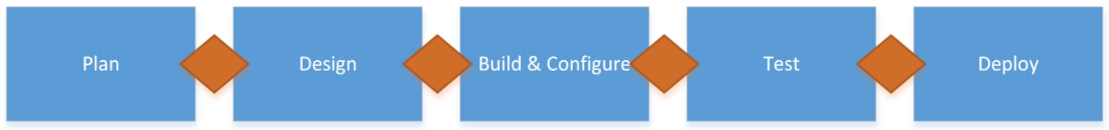
There are traditionally three types of releases (Howard D., 2010):

- Minor release: Consists of small enhancements and fixes; some minor releases may have been completed as emergency releases. Minor releases will supersede all previous emergency releases. More frequently done than major releases, minor releases can range from once a month to quarterly.
- Major release: Contains significant or large portions of new functionality, major upgrades, or new service implementations. Major releases will supersede all minor and emergency releases. Less frequent and requires significant planning and lead times. These are usually done only once a year.
- Emergency release: This type of release is done in response to an incident or significant problem and may be related to the emergency change process. Typically the release is small and similar in nature to a minor release, but is completed in much less time.

Release management is traditionally divided into five different stages (see picture 7):

- Plan. The release plan should at least include the scheduling and content (what changes, enhancements and patches the release contains) definition. Other components might be testing plan, communication plan and governance definition. It is also good to add risk management at this stage. The plan has to be approved before the process can be continued to stage 2.

- Design. The actual release is technically planned in this stage. This includes the software and hardware components. The design has to be approved before process can be continued to stage 3. This is also an approval to start the work in the development environment.
- Build and configure. The actual creation of the release is done in the development environment. Unit tests are run in this stage. When it is agreed that the build is ready for testing, the process can proceed to stage 4 and the release can be transferred to the test environment.
- Test. The testing is done by the testing plan. The best possible plan for testing is to let the business test. The unit tests have been done at stage 3, so no logical errors should be present in the release. When the tests are done, a test report is created.
- Deploy. If the tests are passed, approval can be given for production use. The release is then deployed to the production environment and also appropriate trainings and communication is arranged.



Picture 7. The traditional release management process has only five phases. The red diamonds mark control/approval points.

The release management process is supported by management and monitoring processes. These processes make sure that the release or several releases are proceeding as scheduled and planned. The monitoring process is providing reporting capability.

The common roles in release management are release manager and release control board. The release manager is essential. The role is again much like the role of project manager. The release manager is responsible for the schedule and quality of the release. The release control board (RCB) is the steering board of the release. The RCB makes all the decisions. The release manager presents the current status for the basis of the decisions.

Release management encompasses deployment management. It is required for the effective transition of the application into use. Deployment management makes sure that the software and hardware packages are delivered and installed in a planned way. It also makes sure that the required documentation, communication and trainings are given.

3.4. Key Performance Indicators

The Key Performance Indicators (KPIs) are determined usually separately for every application. They are the tools for interpreting the operation of the application for improvement use. The KPIs can be about quantities or quality. They can be defined separately for each AM service or even separately of each application.

The most common KPI for application management is the SLA, Service Level Agreement. The SLA denotes the agreement of the times the service should be delivered. For instance the different types of incidents might have different SLA definition. Two most used SLAs are the response time and resolution time. Response time tells that if the incident has been taken into the process in time. Resolution time tells that was the incident resolved in time. In addition to these, a SLA could be set for the fact that how many tickets have been reopened. SLA is a KPI that is measured by quantities.

Average resolution time, availability, performance or average hourly rate are few good examples of other KPIs. The different KPIs might have different motivators: performance is about usability of the application, average hourly rate is about financial performance and average resolution time is about the performance of the AM service.

The data collected to support these objectives must measure directly, or provide the ability to derive, the end-to-end service availability and responsiveness experienced by end users.

The best KPIs can be used to improve the application and the service. The KPIs should be used to detect anomalies and provide sound data for the analysis, which should lead to improvement actions. (Sturm R., Morris W., Jander M., 2000)

To tie the application management more to the business processes, so called business KPIs can be determined. We could for instance define a KPI for a logistics company, which measures how many units are shipped per hour. Then some adjustments could be made to the application and the same thing is evaluated again. Business KPIs should bound also the business. This way the corporate IT and the business could have a common goal.

3.5. Continuous Improvement

When an application management service is set up, a strategy and then a design is defined for it. As business needs change, the initial setup cannot used for the whole service lifecycle. The service has to be continuously improved.

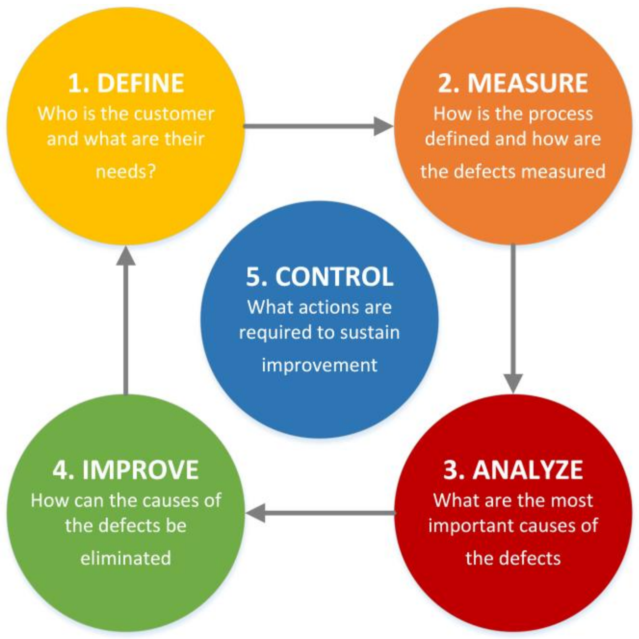
The most effective continuous improvement model follows the Deming's circle (Arveson P., 1998). The idea of the cycle is to chain the activities to continuously running loop of plan, do, check and act: first plan, then execute the plan, check the results and act based on the results. Then plan again. The Deming's circle has been a popular basis for more detailed models. It is a simple, but very effective and poignant model.

A continuous improvement plan can be defined for the service in the design phase (OGC, 2007). The plan describes how the service is measured (the KPIs) and how the results are used to improve the service. This follows the Deming's circle quite flawlessly.

Integrating Six Sigma to ITIL continuous service improvement practices has also been suggested (Harmon P., 2007). Six Sigma (see picture 8) is a well-established and globally followed process improvement methodology. It is particularly compatible with ITIL. Six Sigma is a quality technique developed and introduced at Motorola in 1986 to identify and eliminate defects in manufacturing processes. The methodology relies on the fact that processes can be measured, analyzed, controlled and improved. (Harmon P., 2007)

The Six Sigma analytic techniques are statistically based. The improvement efforts of this methodology are based on the Voice of the Customer (VOC). This means that the quality improvement efforts address only the quality issues that impact the customer. The methodology is used to identify, reduce and potentially eliminate process variances or poor performance that create errors impacting the customer. (Harmon P., 2007)

Six Sigma will be referenced later in chapter 5.2. when defining the release management process. As the process will be for customer facing use, Six Sigma is a perfect methodology to use in the definition work.


1. DEFINE
Who is the customer and what are their needs?
2. MEASURE
How is the process defined and how are the defects measured
5. CONTROL
What actions are required to sustain improvement
4. IMPROVE
How can the causes of the defects be eliminated
3. ANALYZE
What are the most important causes of the defects

Picture 8. The Six Sigma improvement process (DMAIC) is a variation or a development of the Deming Cycle.

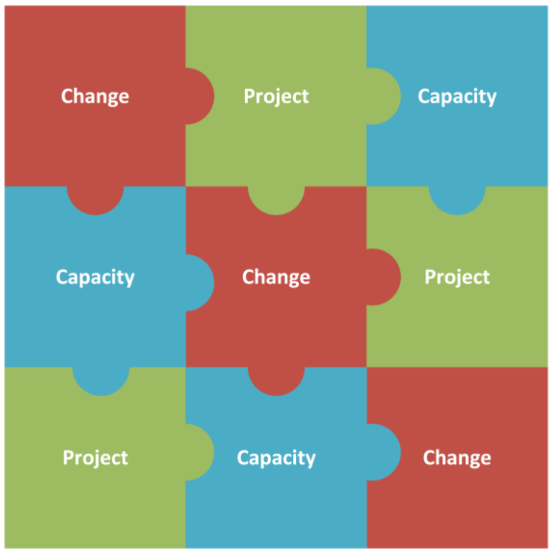
4. Emphasizing the Role of Release Management

The focus of release management is the protection of the live environment and its services through the use of formal procedures and checks. Release management will ensure that changes to the production environment have a minimal impact on delivery of IT services.

- OGC, 2007

Release management is like building a puzzle (see picture 9). It is crucial at managing enterprise applications correctly. Its scope should cover all of the applications in the landscape.

Only then we can discuss about really having a holistic grasp of the whole domain. In this chapter we concentrate in the concept in more detail: what is release management, why should it be used and what happens when it is not used, how it relates to other AM processes and what processes can we introduce to enhance it. Release management is a part of the ITIL transition phase of the service lifecycle. This is reasonable as release is a tool to transition the development work into production use and into the scope of the continuous support processes.



Picture 9. Release management is like building a puzzle. A release can be a collection of different objects, which have to be combined as a coherent whole.

4.1. Definition of a Release and Release Levels

There is no common definition for a release. ITIL defines it this way (OGC, 2007):

A collection of new and/or changed application objects, which are tested and introduced to the live environment together.

The ASL equivalent for release is a change package. It is defined in the following way (ASL Foundation, 2004):

The change package is the set of objects, which have been changed and approved and which will be transferred to the production environment.

Three terms are included in both which define a release: collection/set changed objects, tested/approved and introduced/transferred to production. A release introduces a selected set of implemented changes, which are tested and approved to be transferred to production. In other words, the release has content, there is control of what is released and the intent is to take it to production use.

The objects that create the content of the release should be identified as configuration items. This item can be a software of hardware component. Release management ensures that the changes done to these items are registered in the configuration management process. This enables traceability and reduces the risk of unseen errors in this and the future releases.

The release has a type; it can be an emergency, minor or scheduled major release. The following factors should be taken into account when deciding the appropriate level for release (OGC, 2007):

- The ease and amount of change necessary to release and deploy a release unit
- The amount of resources and time needed to build, test, distribute and implement a release unit

- The complexity of interfaces between the proposed unit and the rest of the services and IT infrastructure
- The storage available in the build, test, distribution and live environments.

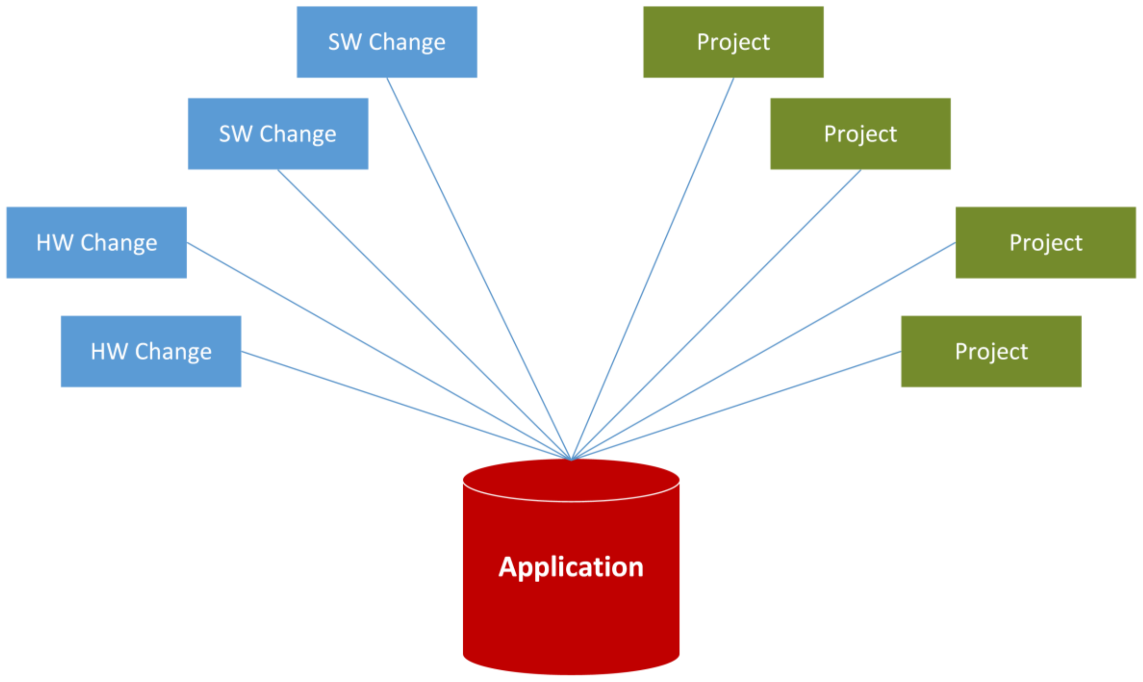
An emergency release is used for releasing changes to production, which fix critical errors. These incidents should have been defined to be priority one or two. Usually the release contains only the changed items that will resolve the current critical error in the production system. This kind of release will be introduced to the system with an accelerated process.

A maintenance release is used to introduce small changed items to the production system. These releases might include small patches or fixes to lower criticality level incidents. A maintenance release is scheduled and there might be several instances defined per year.

A scheduled major release is usually determined to happen a couple of times per year. It will include a set of changed configuration items, which will be introduced into production. These can include the work done in one or several projects and larger changes. These changes are usually new features to the system or some fixes to non-critical issues in the system. Testing takes the most time and effort when implementing a major release.

4.2. Release Management or No Release Management?

An enterprise application without release management plan could be described a world in chaos (see picture 10). When an organization is highly dependent of IT, a single small error can have huge effect to the business performance. Without release management more of these errors are introduced into the systems because changes to configuration items are done in an ad hoc and unplanned manner. The introduction of the changes can then be done without any regard to the dependencies the systems have or what other effects the changes might have on the operation.

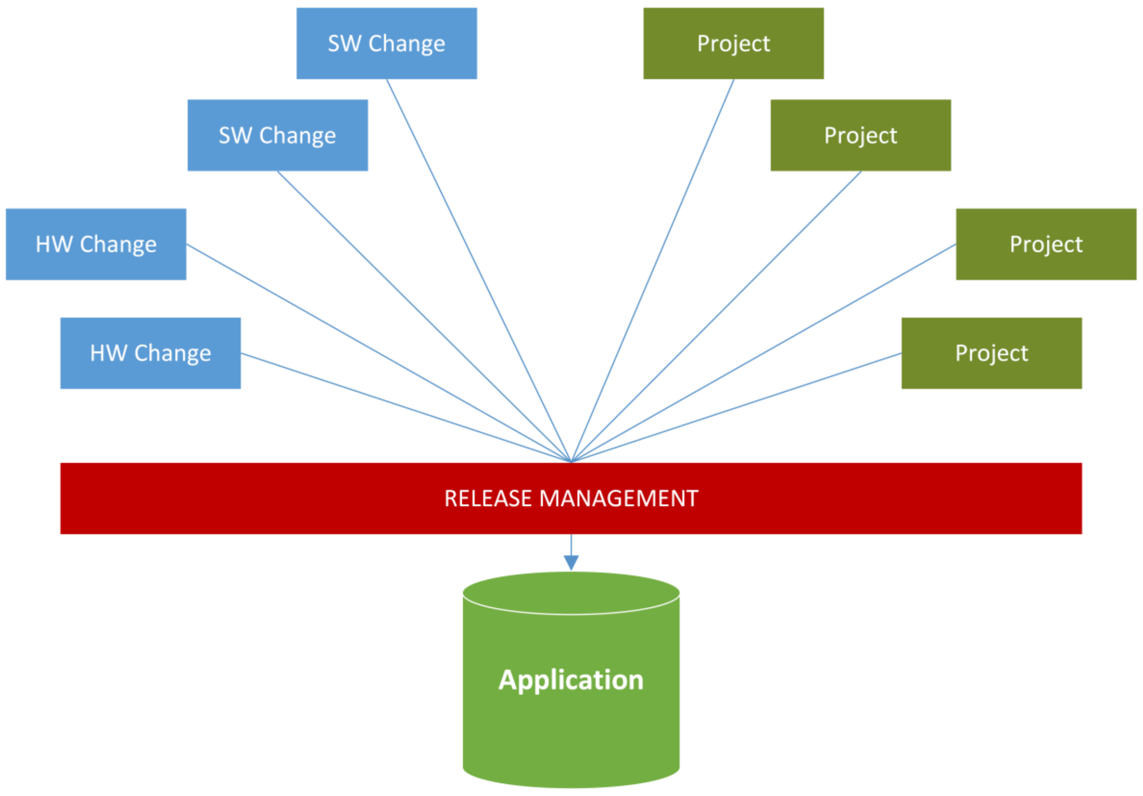


Picture 10. A bad situation: changes to the application are introduced without any centralized control. Application takes all the stress and risks.

The above is an overstatement but something of a worst case scenario. Release management brings a control to introducing application development work to production systems. If a release management process is in effect, no changed items can be introduced bypassing the process.

Release management guarantees that the introduction is done right (see picture 11). Based on the level of the release the activities required may vary. But release management makes sure that the relevant people are informed about the introduction of the items to the system. It also makes sure that there is always somebody responsible. Documentation and training are done before the release management process is done.

The process is also self-learning. The post release reviews provide a tool to reflect upon what has been done and find the elements for improvement. These improvements can be done for the next or parallel releases. This way the organization is gradually working in the correct mode and release management is introducing the quality and cost benefits it should.



Picture 11. Better situation: Release management controls what is taken into production, when and how. Release management takes the stress and handles the risks.

Release management brings more visibility to the future and so gives more ability to act proactively. Release management helps the enterprise IT to know how a particular application is going to change and live in the near future. It provides information based on which the proper enablers can be built and enabled before the release into production use. These enablers can be resources or technical capabilities.

Planning the release management strategy and policy gives more ground to the other areas in IT planning. They define

- What kind of releases are in use
- How the releases are scheduled
- What content the different releases can have
- What is the basic structure of each release type

- What are the release process roles
- What is the approval criteria for different stages of the process
- Who has the approval rights
- How the process is continuously improved

This way the strategy defines what the release is in the organization (content), who is working with the releases (roles) and how the releases are operated (policy). The process improvement factor should take an external view to release management. For this purpose, proper monitoring and management processes should be in place.

Every organization should also plan the scope of the release management process that it would suit the operations. Too much overhead in the process should be avoided or else the process utilization level might not be high enough. Then the benefits of the process would not realize.

Smaller organizations can live without release management but for enterprises it is a must in some form or another. Utilizing only change management is enough for smaller organizations. It has also similar control structure as release management has but it does not take into consideration the the larger scope.

4.3. Release Management in More Detail

A release can be described as a project with a fragmented scope. This also means that a release is basically handled like a project depending on the size of the release. As detailed earlier, releases can be categorized to major releases, minor releases and emergency releases. All of these require different processing, planning and management. They all have a totally different purpose. Minor and emergency releases do not usually require a project like approach.

The differentiator between release management and project management is their primary objectives: release management has the objective of delivery of the entire lifecycle of the service, and the objective of project management is the delivery of the agreed-upon requirements of the approved project within the parameters of time, cost, and quality. While both have the objective of quality delivery, many times project management may sacrifice quality to deliver the respective project on time and within budget. Release management, on the other hand, is concerned with the quality and operation of the overall service from inception to retirement of the service. (Howard D., 2010)

A release manager is basically a project manager and should have similar attributes and competences. The release manager is responsible for the schedule and financials of a release. The release manager makes sure that the release team is built of the right number of competent resources to keep the release on track.

The only difference is that the release might not have unified content. One release can have content from different IT streams: from projects or change management. Release can also be devoted only for a single project. Then the project manager is the release manager. The content should then be harmonized so that the configuration items introduced to production are part of a set of features. This actually applies to all releases: if the content is harmonized, the release is easier to handle and the test and approval processes are more fluid.

A release has to be planned. Release plan is the main deliverable of release management after the actual release. The starting point for release planning is different when considering waterfall or agile approach. The agile release plan has to be planned in iterations as the more traditional waterfall model is planned at the start of every release. In the following we consider only the traditional model.

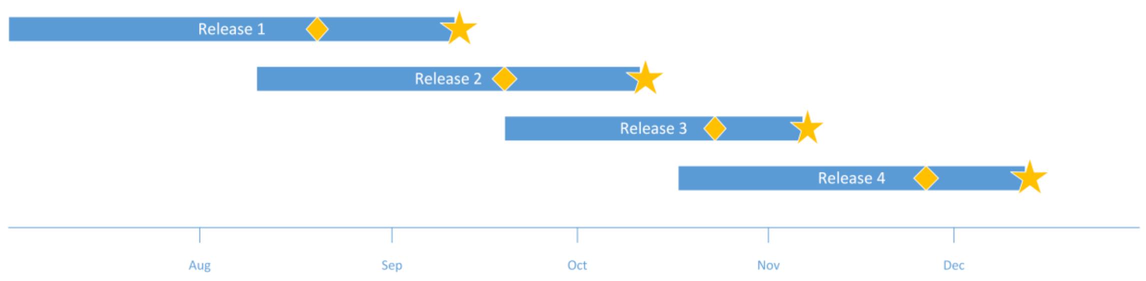
The scope of the release plan depends on the size of the release. Long planning should be avoided with emergency releases and also a more limited plan should be done for maintenance releases. The content of the plan should be determined separately for each release but a minimum set of content should be defined in the release strategy. A release plan could hold the following content:

- Schedule, including the deadline for submitting new content for the release. The schedule should contain also the control points in which the approvals are made.
- Initial content, consisting of configuration items. The release should have initial content planned, but also additional content can be submitted to a certain date. The content should be sized accordingly to the schedule.
- The organization, which will implement the release. This organization might be a single person, a single team or several teams. The main purpose here is to describe the organization and the possible dependencies.
- Change management plan. There has to be a plan based on which the release manager knows how to react to different changes to the content and schedule of the release.
- Risk management plan. The release plan has to describe what the most possible risks are and what their impact to the release would be. Also the mitigation actions should be described.
- Test plan. This plan will describe how detailed the testing should be and what are the main targets. Also the test audience should be listed.
- Approval criteria and governance model. Every release plan should contain the criteria based on which the approvals are made. This policy is applied on the control points. The governance model details the approving organization and how it will be communicating.
- Communications plan. The communications plan will list out when, what and how will be communicated. This plan is then utilized for example when the release is deployed in to the production system.

- Training plan. Release management is without effect if the release is not utilized in the appropriate manner. Training has to be given to the relevant people. In scope are the users but also the people who will maintain the system and support the users.
- Deployment plan. This part of the plan describes how the release is taken into the production system. The description includes who are involved, what should be done and when. A back out plan has to be included so that when there are problems the changes can be reverted.
- Reporting and monitoring plan. How the whole process is seen externally is done by reporting and monitoring. The reporting is used for steering the whole release.

Creating a release roadmap (see picture 12 for an example) enhances the process and its benefits to other processes. The release roadmap is a plan, which is above the individual release plans. In the roadmap, the sequence and scheduling of different releases can be planned on a higher level. This way the dependencies can be seen and drawn more precisely. Different content can be prioritized and placed in release based on this prioritization. The roadmap is a living plan, which means that it also affects the content of the releases constantly.

The roadmap brings more visibility to the work done in application management and projects. There is more time to plan and react to the different needs that can rise from the release work. This has effects to capacity, resource, continuity and portfolio management. When the roadmap is clear for the next couple of years, there should not be so many sudden needs. The capabilities and resources are used whenever and wherever they are required and it is even easier to satisfy any sudden needs. Release road mapping is in this sense very powerful tool, which enhances the use of resources and introduces cost and quality benefits.



Picture 12. A release roadmap can be drawn as a timeline. In this example the release deployment is marked with a star and the deadline for the release to be ready for testing with a diamond.

4.4. Release Management in Relation to Other IT Processes

Release management is coupled most tightly with the change management process (Klosterboer L., 2009). The change management process provides content to the release management process. Release management controls change management so that the changes are done the right time and in the right way. Release management is responsible that the changes implemented are not against any corporate quality policy and they do not risk the production use of the system. Change management is concerned of the short-term view and release management of the long-term view (Klosterboer L., 2009).

The relationship of release, change and project management can sometimes be a bit unclear. As change management, projects also input content to release management. They both introduce changed configuration items. But release management does not control project management. This maybe depends on the nature of the project: if it is traditional or agile. Traditional projects usually introduce contents to release management only in the end phase of their lifecycle so usually they have a devoted release.

The agile projects however are continuously integrating new changed configuration items to production so they utilize the best suited release. If defined so in the release strategy, an agile development project might have its own release schedule where a release is done at the end of every development iteration. This might mean a release for example every three weeks. Agile development is though for developing new solutions, so in application management context it is not so relevant.

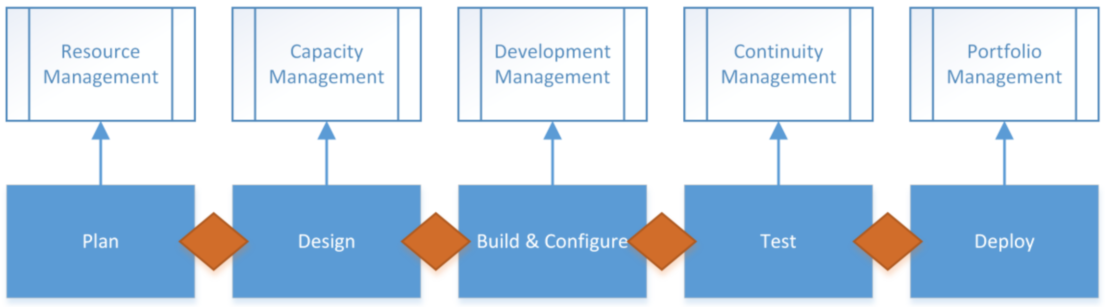
As previously stated, release management affects centrally also the following processes (see picture 13):

- Capacity management. This is regarding the IT capacity, in particular the technical capacity. A release might have preconditions about the capacity: good example is that the changed configuration items might require more physical memory or processing power for the servers.

- Resource management. A release requires resources during its lifecycle. Though the need might vary during the lifecycle. Resource management should be able to provide the resources required to handle the release. Release road mapping should work closely with resource management to plan resourcing in long term.

- Continuity management. Continuity management is concerned about the business continuity. When a system is supporting a business process, its use should have continuity. Continuity management makes sure that the enablers for the business use are in place. This is linked to capacity management and release management.

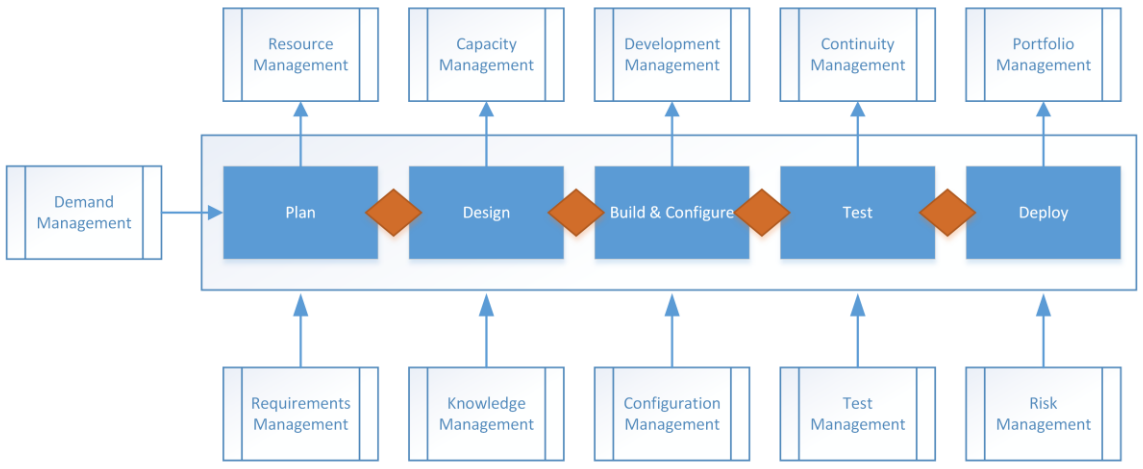
- Portfolio management. Release management provides new content to portfolio management. The release introduces new features to the system, which should be noted also in portfolio management.



Picture 13. Release management affects many other processes. When the release management process is well defined and working as should, the other processes have a better chance to react to the needs of the release management process.

4.5. Supporting Processes

Release management is dependent on various other processes (see picture 14). Incidents, problems and projects generate the input to change and release management. Release management controls the change management process. Demand, requirements and risk management are enabling processes for Release Management. Test, configuration and knowledge management are supporting processes, which are utilized by release management. They are in chronological order demand, requirements, risk, test, configuration and knowledge management.



Picture 14. The release management process is also dependant of other processes, which are required to implement the release in the best possible way.

4.5.1. Demand Management

Demand management is at the interface of the customer and the release management process. The key principle of demand management is that never take more work than you can handle. In other words, demand management makes sure that there is enough capability in release management. If it is possible to raise the capacity, demand management raises it to consideration.

Demand management is tied together with capacity and resource management. These together form the release capability. Release management requires resources to handle releases. The releases require technical capacity to be usable as intended. If these are not available, there is no release capability.

Demand management can also be an input to release management. Demand management can pull new requirements from the customer, which can lead into new content to the releases. The most important function for demand management is to know when there is available release capability and when there is not.

4.5.2. Requirements Management

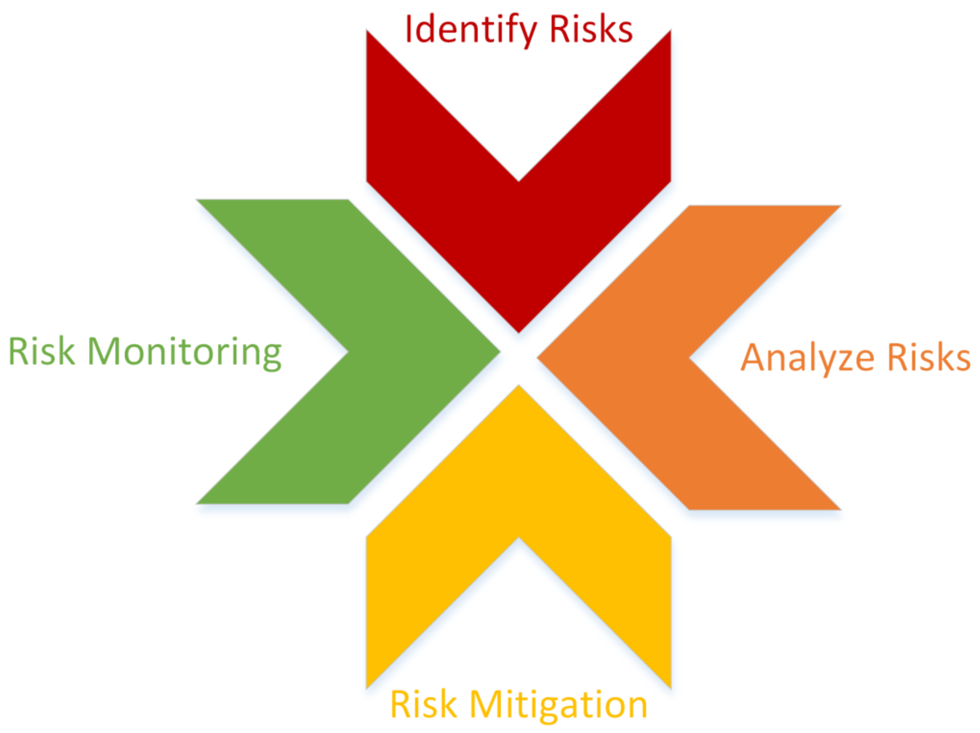
The demand for new applications or features come in different forms. The business might raise some issues in meetings. A defect in the system might be detected in the system and a fix should be created. A new business process is introduced and it requires a supporting IT solution. These different inputs might come to the IT via different channels.

Requirements management makes sure that the actual need is analyzed and understood right. It gathers the needs from different formats and analyses what is actually behind every request. A decision can then be made if the requirement is valid or not. Sometimes a requirement is a duplicate of another requirement. Sometimes it can be identified that a set of requirements can be fulfilled by a single solution. In best cases the requirement can be fulfilled by some already existing feature. So it is very important to know what the business is really aiming at. The worst initial requirement from business is a detailed description of an IT solution.

Best requirements are described as use cases or scenarios. It is easy to see from a use case if there is actually a need to provide a solution for the requirement. Good requirements can also be used as basis for the test cases used later when testing the implemented solutions. In this way requirements management benefits also the latter parts of release management.

4.5.3. Risk Management

One big task that release management has is to mitigate the risks that changes to the applications might cause. Risk management (see picture 15) is the separate process used for this purpose. Corporate IT can have a separate function for risk management.



Picture 15. The risk management process is four phased. The risks have to be identified and analyzed before mitigation actions can be defined. The risks have to be continuously monitored.

The risk management process is a continuous process. So it has to be done constantly also during the release management process. The risks should be first identified and then assessed. The assessment should conclude in the following:

- The risk description. The risk should be described in a manner, which is understandable and also reusable.
- The risk probability. How probable it is that the risk realizes. The probability is usually presented as a percentage.
- The risk impact. This describes that what happens if the risk realizes. The impact might be a quantative or qualitative value.
- The risk mitigation actions. What is done to prevent the risk from realizing?

This information can be used to determine for example that is it reasonable to introduce some changed configuration items to the production system. The change might be rejected if the risk is too high. This assessment information should be also used really to mitigate the risks and also mitigate the impact when the risk is realized.

4.5.4. Test Management

When an implementation is ready it has to be verified that it is ready for production use. The test management process is utilized for this verification. In release management, testing is used to validate the changed configuration objects against the (use case based) test cases and also the criteria (policy) set in the release management strategy.

Test management process starts with creation of a test plan. In release management, the test plan can be defined per release or for all releases. The latter might be a general master test plan, which can be tailored separately for each release. Some release strategies might specify that certain types of tests should be done for each release. The test process is managed by a test manager who should be separate person from the release manager. The test plan should also include the information of how the tests are implemented: in a controlled test environment or somewhere else and who are the testers. There might be a need to use testing professionals but in many cases selecting few of the future end users for testing provides adequate results.

The test plan contains one or more test cases. The best test cases are based on use cases, which are in turn based on user requirements. The test cases describe step by step how different tasks should be processed in the test system. The test cases have some conditions that should be met before the tasks can be started and some conditions for the task to be passed.

The test cases are run in certain sequence. The tester writes down the observations and the result of the test case. If the test case failed, a description should be included about the incident that prevented completing the test case. Depending on the release strategy, a single failed test case can prevent the acceptance of the release to the next phase of the release process.

Unit tests are not a part of the test plan. Unit tests are defined and carried out by each implementer and the corresponding fixes should be done before passing the changed configuration items to the actual business testing. The master plan can contain the required procedures for unit tests.

Separate from the actual testing is also is the user acceptance test (UAT). In the UAT it is determined if the new features are compliant of the actual user requirements delivered and if the release is applicable to the production use or should the release still be mended to suit its purpose better. UAT does not aim to find bugs or other malfunctions from the release. If the UAT result is positive, the release can be accepted by the release control board for production use.

4.5.5. Configuration Management

Configuration management is actively being used in the infrastructure management. In application space it is called Software Configuration Management (SCM). Configuration management is used to provide traceability to the changes targeted to the configuration items (CI). It also provides management of the dependencies between the configuration items. Configuration management is a tool for risk management and development.

Configuration items should be also defined in application configuration management. An application CI can be for instance a code component, a web service or an integration. All of these can have dependencies, which are good to recognize when changing them.

As an engineering discipline, SCM provides a level of support, control, and service to the organization (Keyes J., 2004):

- Support. SCM is a support function in that it supports program engineers and developers, the program, the corporation, and, in a number of situations, the customer.
- Control. SCM is a control function in that it controls specifications, documents, drawings, requirements, tools, software, and other deliverables.
- Service. SCM is a service provider in that it supports people and controls data.

Release management is responsible of updating configuration management of the changed configuration items, both application and infrastructure related. For release management this is very important especially when there is a need to roll back the release if a showstopper is detected in the new functionality despite the comprehensive testing.

4.5.6. Knowledge Management

All of the application management processes produce and use a big amount of data. This data is about the application itself or how the application is maintained. The release management process is responsible for a large part of that data and is responsible for providing it to the knowledge management process. This enables the corporate IT to use the information again.

Managing knowledge is one of the biggest problems that current day organizations have. Much information is stored into individual workstations, laptops and various places in the enterprise network space. The information is not structured properly and it is hard to find. One challenge in particular is great: big part of the organizational information is stored in the people that are part of that organization. This is called organizational memory (Jennex M.E., 2007).

Organizational memory can be viewed as abstract or concrete. It is comprised of unstructured concepts and information that exist in the organization's culture and the minds of its members, and can be partially represented by concrete/physical memory aids such as databases. It also is comprised of structured concepts and information that can be represented exactly by computerized records and files. (Jennex M.E., 2007)

When the people leave the organization, they take this silent information with them if it is not actively collected and stored elsewhere. There should be a way to store that information in proper form and at the right time. One way is to interface knowledge management to other processes and gather the information silently. That way knowledge management would not be a separate process presenting only overhead.

The main aim of knowledge management is to make the information available in reusable form. Thomas H. Davenport and Laurance Prusak (1998) call this knowledge codification. It creates some quality requirements for the information that release management produces. The information stored should be produced in an understandable form. The language should be good and diagrams, pictures and charts should be used whenever applicable. Even modern day multimedia could be used. Storing knowledge should be a natural part of the actual process. For example an expert system represents an explicit attempt to capture or imitate human knowledge by transferring it to a formalized rules-based system (Davenport T.H., Prusak L., 1998).

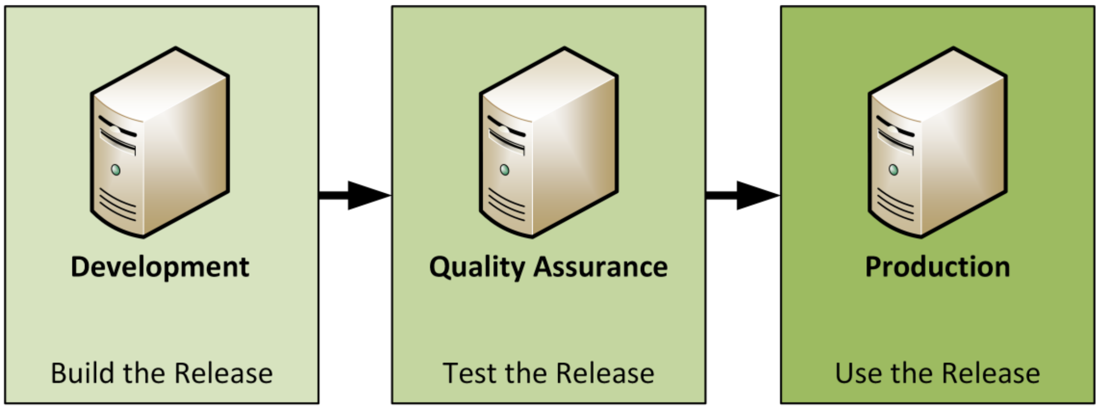
The release management should provide at least the following information for knowledge management purposes:

- Test plan and results
- Changed system documentation
- Release management process observations
- Memos of meetings

This information should be actively stored. As said, some information can be gathered also silently. This should be integrated to the processes and the best way to do that is to integrate to the tools, which guide the processes. There are many different tools for knowledge management purposes. Best way of gathering the data is to store it to the process tool, for example in ERP or CRM. The use of social networking is growing in enterprises, it can be also utilized in information gathering. There are also more traditional knowledge management tools as corporate document systems, workspaces and dedicated knowledge management systems. The main knowledge management system is called the Knowledge Management Database (KMDB).

4.6. Release Management Technical Environment

The aim of release management is to introduce changed or new configuration items to the production systems. But before that is done, the release has to be implemented and tested. There are also dedicated environments (see picture 16) for this.



Picture 16. The release management environment. Each separate part of the environment has a defined purpose.

4.6.1. Building the Release

When there is only the aim to test a solution, a sandbox environment can be used. The actual development should be done after thorough planning in the dedicated development environment. This environment does not have any production use and the data can also be anything. This usually makes managing development easier as the security clearances do not have to be so high as the data is not actual business data. Unit tests are done in the development environment to assure that there are no logical defects in the release.

4.6.2. Testing the Release

When the actual implementation is approved by the release management board, the release can be transferred to the test or quality assurance environment. The test environment is as close to the production environment as possible. This is made sure by copying the production data to the test system periodically. The only thing that separates the test system from the production system should be the changed configuration items presented by the release. The tests are conducted according to the test plan in the test environment as is the user acceptance test.

4.6.3. Deploying the Release

When the release is approved to production use by the release control board, it can be deployed. The production system is reserved only for production use. This means that the system is used for supporting or driving business processes. No changes should be done to the production system before the changes are tested in the test system. This means that the changes should be always implemented in the development system from which they are transferred for testing in the test system. Production system usually has more technical capacity to support more users with the required performance.

4.7. Challenges and Problems in Release Management Process

Antti Lahtela and Marko Jäntti listed the challenges and problems in release management in their case study (2011):

1. There is no specified release management process.
2. The release manager role is unclear.
3. The customer does not know what the release contains.
4. The release distribution rate is too high.
5. The customer thinks that the service provider cannot test or inspect all the test cases.
6. There should be more test environments in the test process.
7. The change management of the test environments is insufficient.
8. Problems in version management.
9. The case organization does not have any specified “release jury”.

In a previous study, Jäntti presents also couple more problems with Hanna-Miina Sihvonen (2010):

10. Informing and communication is lacking.

11. Releases have quality problems.
12. Lack of metrics in the release management process.

These problems tell the simple fact that release management is often neglected or not well-defined. Jäntti and Lahtela (2010) urge organizations to define the release management process, the roles in it and also communicate the definitions to all relevant process stakeholders. It is most important for the customer of the process to understand the principles.

The problems listed are among the ones that will be addressed in chapter 6 of this thesis when the release management process is defined. The process will differ from the one presented by Jäntti and Lahtela (2011), whose process I saw to be too lightly defined in the paper.

5. Process Development

Before we continue to presenting the actual process definition case, it is good to go through some process modelling techniques. To do this we should first discuss about what are characteristics of a process.

5.1. What is a Process?

Release management is a part of a large IT process family. In nature all processes are an everyday phenomenon. They occur all around us. Some processes just go naturally but some need nurturing. Process as a word can mean different things. It has many different definitions, which depend on the point of view from which it is examined.

The basis of the whole word process is movement. Things are naturally in motion or have been set in motion. And things follow other things. Results are produced.

Processes require resources, they are not perpetual-motion machines. They need expertise to guide them. Processes produce changes. And those changes might affect other processes or even the process at hand. Then, if we would describe a business process, it would sound something like this (Fyrvik T., Uthaug E.S., 2005):

A business process is normally defined as a chain of logical connected, repetitive activities that utilizes the enterprise's resources to refine an object (physical or mental) for the purpose of achieving specified measurable results or products for internal or external customers.

Processes usually have driving forces, like customers. In a healthy process the chained activities are logically connected. The end result can be physical or mental. In a business sense it can mean a product or a service. In itself, a service is a process, isn't it? In IT the definition of process is somewhat a combination of these two descriptions above.

The process has a sequence, it needs resources, the result is some physical or mental object, which should be measurable and it is usually produced for some customer. Basically it is about people doing their work with tools supplied to them. This tool might be a wrench, pen and paper or most interesting to us, an application. This work is done because somebody needs the end result for their own purposes, which may be for example to utilize it in a process of their own.

Processes themselves are always in the process of evolution. Some processes are that more than others. They are never quite static. The evolution might be naturally inflicted from within the process or it is developed by external actors. Monitoring the process and the end result can give the best advice for the development. Sometimes the development need is presented by the customer. Business wise the best development is delivered through innovation. When you discover totally new ways of doing things, the more advantage you have over your competitors. More subtle ways of changing the process give also an added benefit.

The development should never stop. There is always room for some evolution. But if the process is not cared for, just the opposite can occur: a revolution might break out and the process could be ended altogether. Sometimes this might be the best thing to do. Sometimes you should just let go of old habits and take up new ones. But the long lasting processes are stable and are reliable. So the goal should always be that the process should survive as long as possible.

The quality of the process is measured by how well the process meets the expectations. If the process structure of an organization is clearly described, it will show (Fyrvik T., Uthaug E.S., 2005):

- What has to be done
- What the expected result is
- How we measure if the process deliver the expected results
- How the results of one process affect those of another process

In my thesis these topics are explored and discussed in the light release management. They are very viable in that context as release management has a definitive goal and it is known how to reach it, the process has to be measured and it has many effects on other processes, IT and business alike.

5.2. Modelling Methods

5.2.1. Work Flow

Work Flow defines how work is broken up across people and how people coordinate to ensure the whole job gets done (Beyer H., Holtzblatt K., 1996). The work flow models can have eight (8) different components: individuals, responsibilities, groups, flow, artefacts, communication, places and breakdowns. The model can provide the answers for example to the following questions (Beyer H., Holtzblatt K., 1996):

- How do job responsibilities get assigned to people?
- What are the different roles people take on to get work done?
- How do new tasks get passed to a person?
- Who do they get help from?
- Who do they have to work with to accomplish their tasks?
- How do they use physical places and artefacts to help them coordinate?
- Who do they give the results to and in what form?

The work flow model is used to give a top level view to the release management process. The process is straightforward at the first look but contains many branches that go and interconnect in many levels. The flow model provides some clarity to the whole process. The questions listed above need all to be answered when modelling this process.

5.2.2. Sequence Models

Sequence models (Beyer H., Holtzblatt K., 1996) describe the steps by which work is done, the triggers that kick of a set of steps and the intents are being accomplished. This modelling technique can be quite low-level and gives a clear view to how the work is really done and how it is structured. Sequences cut across the other models, tying them together. Sequences are time-ordered. They show how different roles interact in different places, using artefacts to support communication and actions to get the work done.

Sequence models are used here to do just what was described above, to tie the other models together and show the interactions between different objects, actors and environments. At the same time in the lower level of granularity the way of the real work should be discovered. A sequence model is a very clear way to represent work as it shows the distinct steps to get the result of the work.

5.2.3. Cultural Models

Cultural model (Beyer H., Holtzblatt K., 1996) is a presentation of the cultural network, which affects how the work is done and the choices the workers make. Every workplace has its culture. The bigger the organization is, the more diverse its culture can be. A culture defines expectations, desires, policies, values and the whole approach people take to their work. Cultural issues are not something physical or technical, so they are really hard to see. They are revealed by subtle things in the way people talk, how they work and their overall attitude. The cultural climate contains the formal and as well as the informal policies of the organization. In a cultural model influencers (people, organizations and groups) are represented. It shows how they influence each other. These models do not map to organization charts. They show how power is experienced by people, rather than the formal power of the organization.

5.2.4. Physical Models

The Physical models (Beyer H., Holtzblatt K., 1996) take on account the physical surroundings where the work is done. The workplace can give many hints about the work itself. The physical environment affects how work is done in every scale. Different to the cultural models, the physical world is easy to see. A physical model is a drawing of those aspects of the workplace that are related to the focus of the modelling. It is annotated to show how the space is used and to show strategies, intents, and cultural values that are revealed by the way space is used. The physical model shows how people respond to the environment by restructuring it. A physical model is not a floor plan or an inventory. It does not show all the objects of a workplace. It is only concerned about those objects, which are related to the work that is modelled.

5.3. Artefacts

Artefacts (Beyer H., Holtzblatt K., 1996) are the things people create or use to do their work. Artefacts reveal the workers' thoughts when they are doing their work. They can be everything from to-do lists to computer programs. Artefacts are manipulated in the sequence models and passed between people in the flow model. They show the conceptual distinctions of the work. An artefact model is a drawing of the artefact complete with handwritten notes, which is extended by the model. Artefacts are the concrete trail left doing the work. They do not work by themselves.

5.4. Combining Techniques for Process Modelling

Release management can be viewed as a social interaction process for the purpose of doing business. The most crucial part of this method is the way how to achieve normative coordination by imposing social constraints and rules in the form of mutual obligations among the actors. Thus it allows actors to talk to each other based on their sets of obliged behaviours and thereby have a clean approach to negotiation. This model can be used to capture, support and enforce social patterns of behaviour of business processes operating in open environments, which inspire trust in the actors.

Combining these modelling techniques is unavoidable and desirable. They all represent a different perspective to how the work can be perceived. They are in many ways interconnected as it has been stated previously. A person plays roles, a role has responsibilities, undertakes tasks and exchanges artefacts with other people to discharge these responsibilities. The methods could be summarized like this:

- The sequence models show how these tasks are accomplished in detail.
- Artefacts are used in accomplishing the tasks.
- The responsibilities and manner of accomplishing them are driven by organizational context and culture as shown on the cultural model.
- The work represented by the sequences is done within the work environment described by the physical model.
- The work is usually driven by obligations and contracts. These are transferred by social interaction in the cultural climate.
- The work is broken up and organised so that people take up the obligations and do their job as coordinated.
- The modelling process itself can be modelled by fine grain modelling and broken to different levels of granularity to point out the problems in the models.

Stepping back and looking at the models together reveals all the different aspects of work and how they relate to each other. It reveals how the whole work of one person hangs together. A mapping between the different methods should be provided to get a more comprehensive view at the process. It gives an advantage to the process modeller. Combining the models might help the modeller to gain a better insight about the nature of the process under study. A good way to start the modelling is to first understand the larger view of the process and then dig deeper to more accurate levels. Depending on the aims of the modelling the level of granularity might be high or low.

6. Light and Adaptable Release Management Process

Based on the descriptions in chapter 4, a light and adaptable release management process will be now introduced. The process is defined for real use but there is no real guarantee that it will be ultimately used. The definition will be based on experience, theory, already available processes and reflections with colleagues. The background for the need and also the development is discussed here, as is the future plan of implementing the process into real use wherever it is required.

6.1. Background

6.1.1. Business Case

My organization provides application management services to its customers. These AM services are a part of a larger portfolio of products and services. The customers are from different industries and of different sizes. So there is much variation on what kind of AM services they require.

This means also that these customers have different kind of approach on release management. Some of the customers have defined and implemented their own release management process. Some of the customers have something like a release management process in use and some have none. To these last two kind of customers we have to provide a release management process of our own.

First reason for this is that release management should be in the portfolio. The second reason for this is that with release management the customer applications can be maintained with a more planned way. If we could introduce an effective release management process, which would suit the most of the customer environments, then the services could be produced in a more efficient and factory like manner. This brings benefits for both the supplier and the customer, mainly the following ones (Howard D., 2009):

- Visibility to the application roadmap
- Predictability to budgeting and scheduling
- Effective mitigation of risks to business performance
- Financial benefits from decreased productive system downtime

The main need is for SAP applications. SAP has published a standard for release management and also supports the process with their Solution Manager tool. SAP is an enterprise application, which is used to drive and support multiple business processes from which many can be business critical. SAP is heavily used in finance, human capital management, logistics and production planning areas among others as well.

Why wouldn't we just follow the standards presented by SAP? SAP has a much defined process how to manage development of new features for their products but it is not perfect. Many enterprises do not utilize the Solution Manager tool properly and require a lighter approach. It has many good features such as integration to the other parts of the SAP module landscape.

The aim is to define a process that would be usable internally and acceptable externally. The process should follow the organization's own internal processes. The IT business has several standards that should be followed to produce acceptable services. This is important also when competing with other similar IT corporations.

The process should suit also the needs of the customers. The concern that many customers might have is that they do not have enough control when release management is outsourced. The introduced process should address this. The customers might also have their own quality and performance requirements, which cannot be bypassed. This stresses more the requirement of adaptability.

While the internal requirements are static, the requirements might vary from customer to customer or even with one customer. The process should also not be targeted to satisfy the needs of a specific industry. This requires more high level approach, which will enable the process to be lighter. It should be noted that too light can lead into not enough control.

6.1.2. Research Approach

The research methods used to gather information for basis of this thesis were all qualitative. The work started in Spring 2012 with peer interviews which were non-structural. The aim of those interviews was to find out the different situations in the various customer accounts regarding release management.

In all, five interviews were conducted. Results of these interviews were used to make sure that the planned release management process would be adaptable for all required services.

Other sources of information were the following:

- Past experience of release management work, including long experience in requirements management.
- Further experiences from the current environment, especially own customer accounts.
- Literature, including books and industry articles.

The earlier incarnations of the process model was submitted to reviews with peers. The process was then reviewed as part of the other application management process family to align all of the processes.

All of this information was analyzed and used in building the process definition described in the following chapters. The analysis was done with the methods described in chapter 5 and also traditional cross-functional flowcharting was utilized to produce a more comprehensive model.

6.2. Release Strategy

When developing new processes, a strategy is required. Actually two strategies should be created: strategy to the overall release management and also the strategies for individual release management relationships. The latter means that a strategy should be defined for each enterprise interfacing to the process. The strategy should contain the definitions of

- Targets: what are the aims of the process
- Means: how these targets are reached
- Policy: what are the ground rules for release management

The targets of the overall process are:

- Bring visibility to the overall change work
- Elevate reusability of work
- Increase organizational knowledge
- Decrease amount of defects introduced into the systems
- Increase customer satisfaction

The means to reach these targets are:

- Enable release management to control deployments
- Link release management to knowledge management
- Link release management to configuration management
- Link change management to release management
- Link release management to test management
- Give customer control on selected parts of the process

The release management policy dictates:

- What are the acceptance criteria
- What does the release plan contain
- How should the releases be tested
- What are the acceptable deviations
- What is to be done when deviations are met

The individual relationship strategies should be defined based on the overall strategy and further defined based on the customer requirements.

6.3. Process Definition

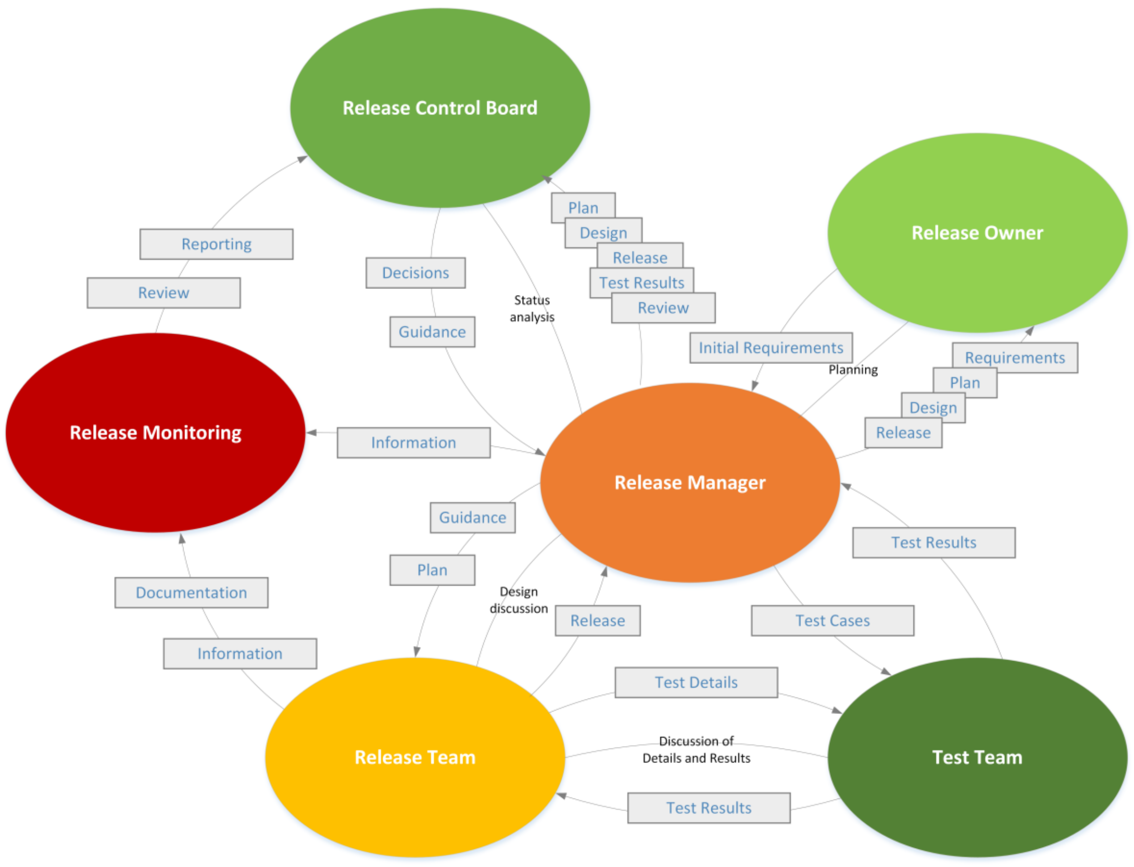
The two main targets for the release management process are that it should be light and adaptable. These targets will be released by following the best possible process creation methods and the models introduced in chapter 5. The approach will be lean and Six Sigma will be referenced.

The Process Flow model introduced in chapter 5.2.1 presents questions. Answering them is a good starting point to this process development task:

- *How do job responsibilities get assigned to people?* The roles are predefined in the Release Management strategy. Roles are manned in the Resource Management process.
- *What are the different roles people take on to get work done?* These roles have been predefined in the Release Management Strategy.
- *How do new tasks get passed to a person?* The new tasks are managed by the Release manager and are passed to different actors in the process by an information system or personally.
- *Who do they get help from?* The actors receive help from the Release Management organization or any stakeholders such as business or other stakeholders like software vendors.

- *Who do they have to work with to accomplish their tasks?* The Release Management team consisting of the Release Manager, the RCB and specialists with different required competences.
- *How do they use physical places and artefacts to help them coordinate?* The process is not dependant of physical places and the artefacts are information systems and IT hardware.
- Who do they give the results to and in what form? The results are passed to the end customer – may that be the business or external customer.

Based on these answers we can already draw the work flow diagram, which is presented in picture 17.



Picture 17. The work flow model for LARMP. The roles are defined in chapter 6.3.2.

As described earlier in chapter 5.2.1, these kinds of processes can have eight (8) different components: individuals, responsibilities, groups, flow, artefacts, communication, places and breakdowns. Apart from places, which is not relevant for modern IT environments, I will go and define these in the following process definition:

- Roles = Individuals, groups and responsibilities
- Process components = Flow and breakdowns
- Configurations Items = artefacts
- Deployment = communication

This process is based on

- ITIL definitions. ITIL has defined a strong reference model for a release management process. This will be the skeleton for the process.
- ASL definitions. Any viewpoints of ASL not addressed in ITIL are taken into consideration and adapted into the process.
- Practical experiences. The writer's own experiences from release management are many and he has also witnessed many successful and also non-successful release implementations. This gives some perspective to the theories used.
- Other industry sources (articles and books). The latest developments in release management have been sought from the latest articles and books available.

6.3.1. Release Management Process Components

Light could be read as lean. All excess overhead has to be cut from the process. There are many things both ITIL and ASL demand from release management but which are really just generating overhead to the process. This should be taken into account in the proposed control points: do they bring any additional value to the process?

Light also means that the process should be easy to understand to follow. When the process is light, there is no room for misinterpretation and so also less room for process error. The procedures in the process should not be overly complex or long and they should have clear inputs and outputs. This makes the process more reachable and also more acceptable. This drives the process to be utilized and the benefits of release management can be better gained.

With adaptable I mean that the process can be

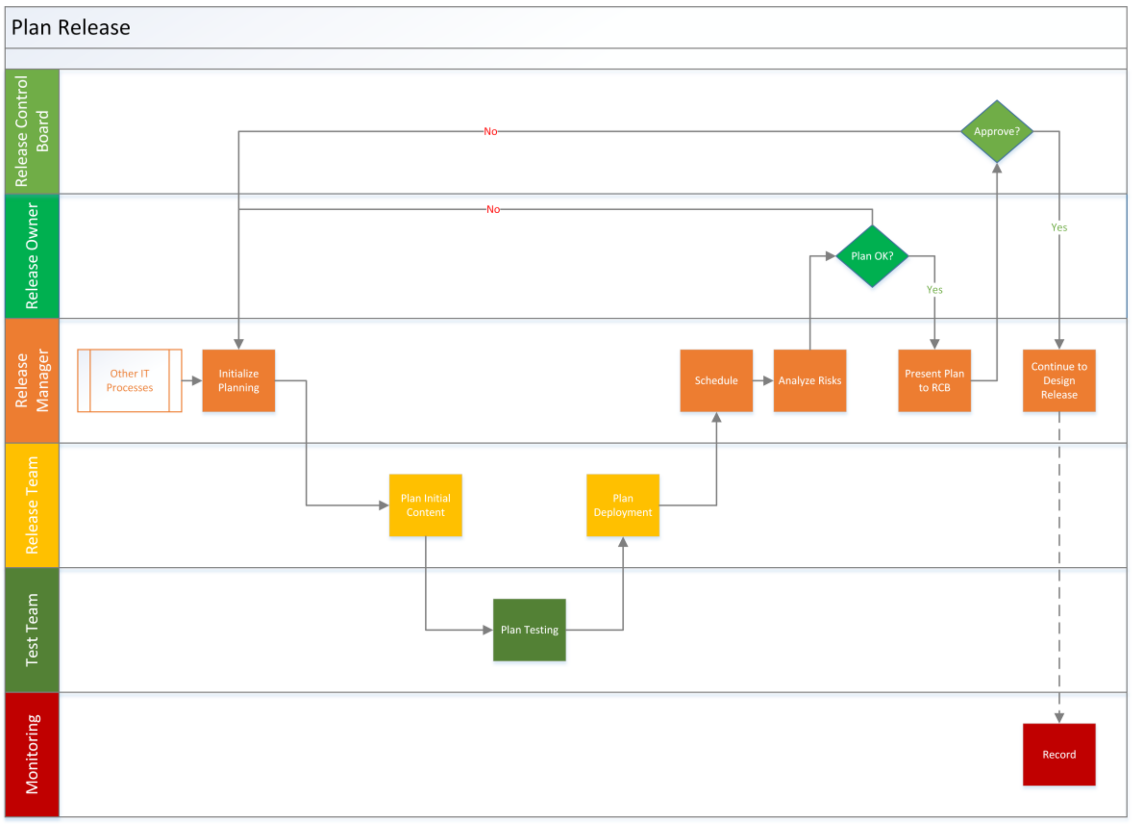
- integrated to other IT processes
- integrated to the customer processes
- integrated to the customer way of working
- adapted to adhere to the customer quality requirements and guidelines
- used with emergency, maintenance and major releases
- changed when the requirements change

With these implemented, the process addresses all the adaptability needs it was set to satisfy. We will use sequence modelling next for the description of release management process sequence. We will call the sequence here as component.

The process components are

1. Plan Release (picture 18). The foundation of a release is laid out in the planning stage. In minimum the contents, test plan, schedule, deployment plan and resourcing should be defined.

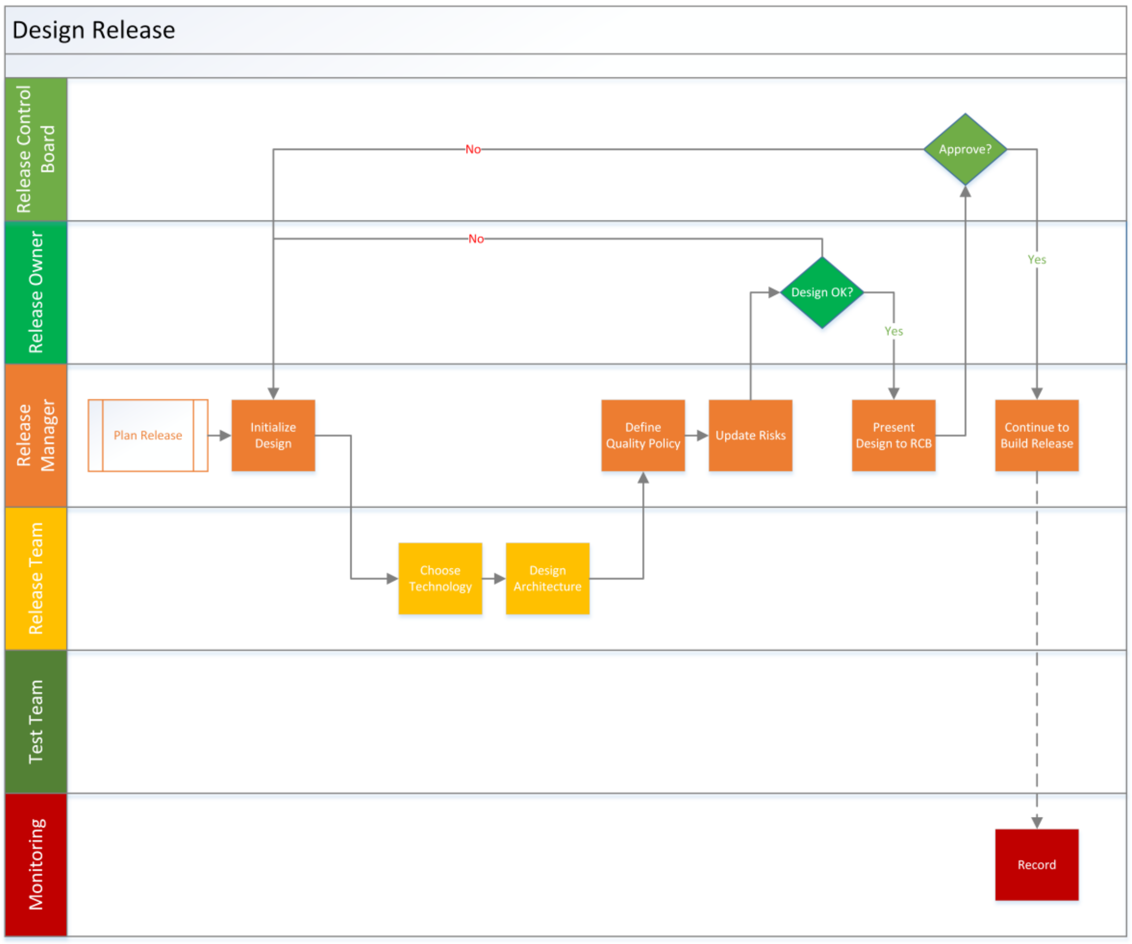
   Acceptance criteria: Plan contains the minimum set of definitions and is coherent.



Picture 18. The release planning phase of the process.

2. Design Release (picture 19). The actual content is designed in this stage. The design should adhere to the standards of the used technology.

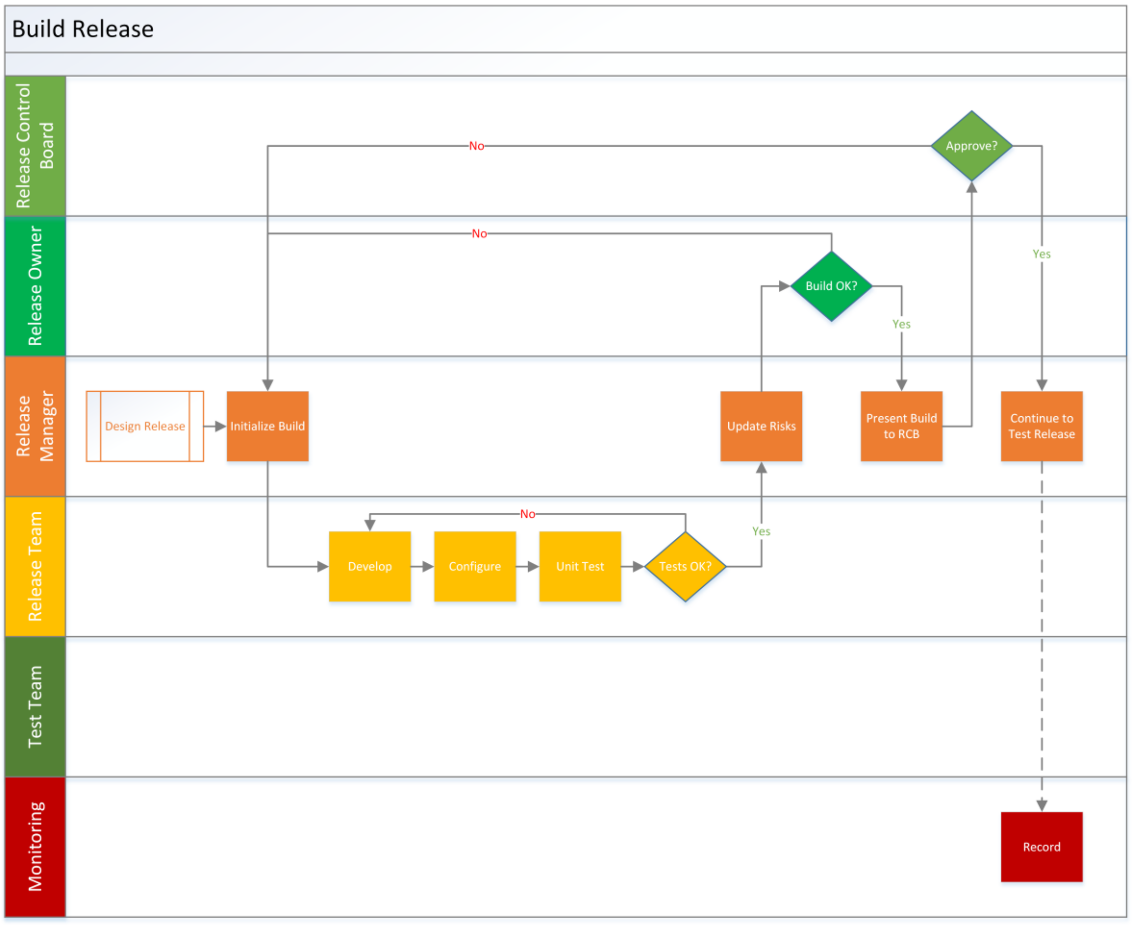
   Acceptance criteria: Adherence to standards and coherence.



Picture 19. The release design phase of the process.

3. Build Release (picture 20). The release build is done utilizing some aspects from the agile development approach: daily meetings, weekly meetings and meetings every two weeks with the release owner. Unit tests are done in this stage.

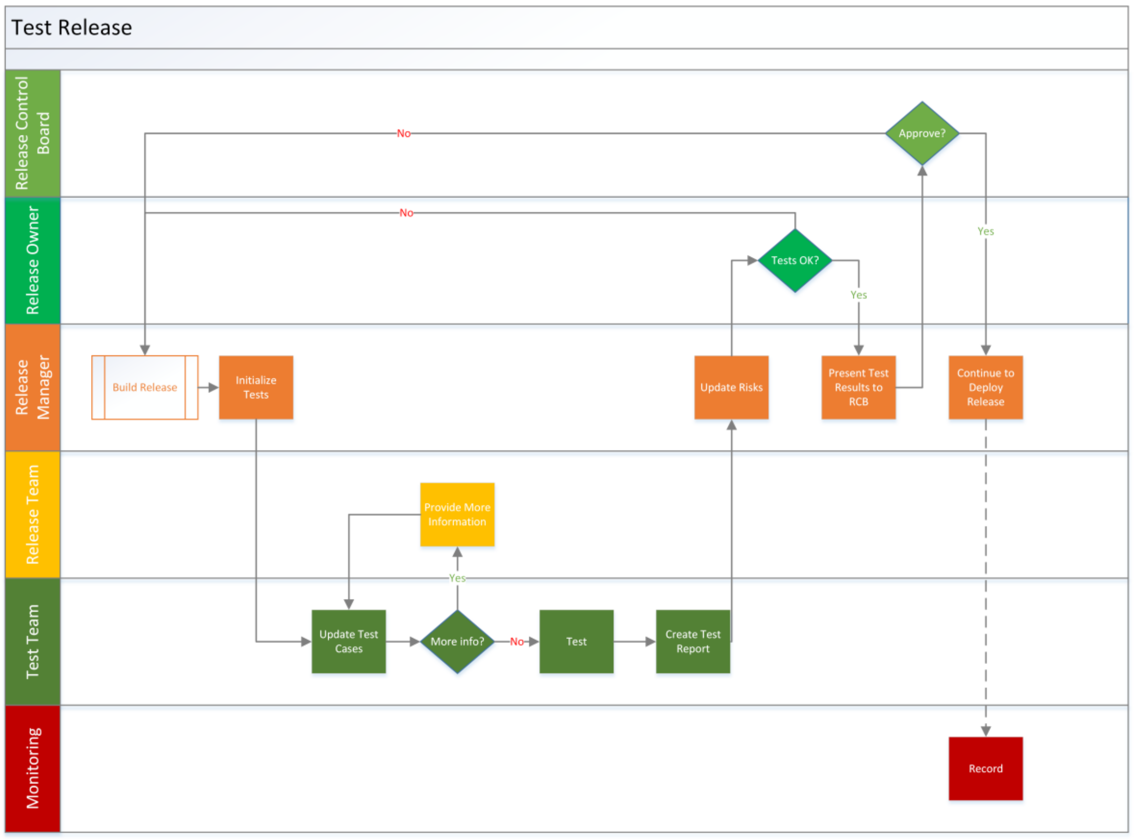
   Acceptance criteria: Unit test results and documentation quality.



Picture 20. The release build phase of the process.

4. Test Release (picture 21). The tests are carried out by following the test plan. When the tests are conducted successfully, the release is sent for user acceptance testing.

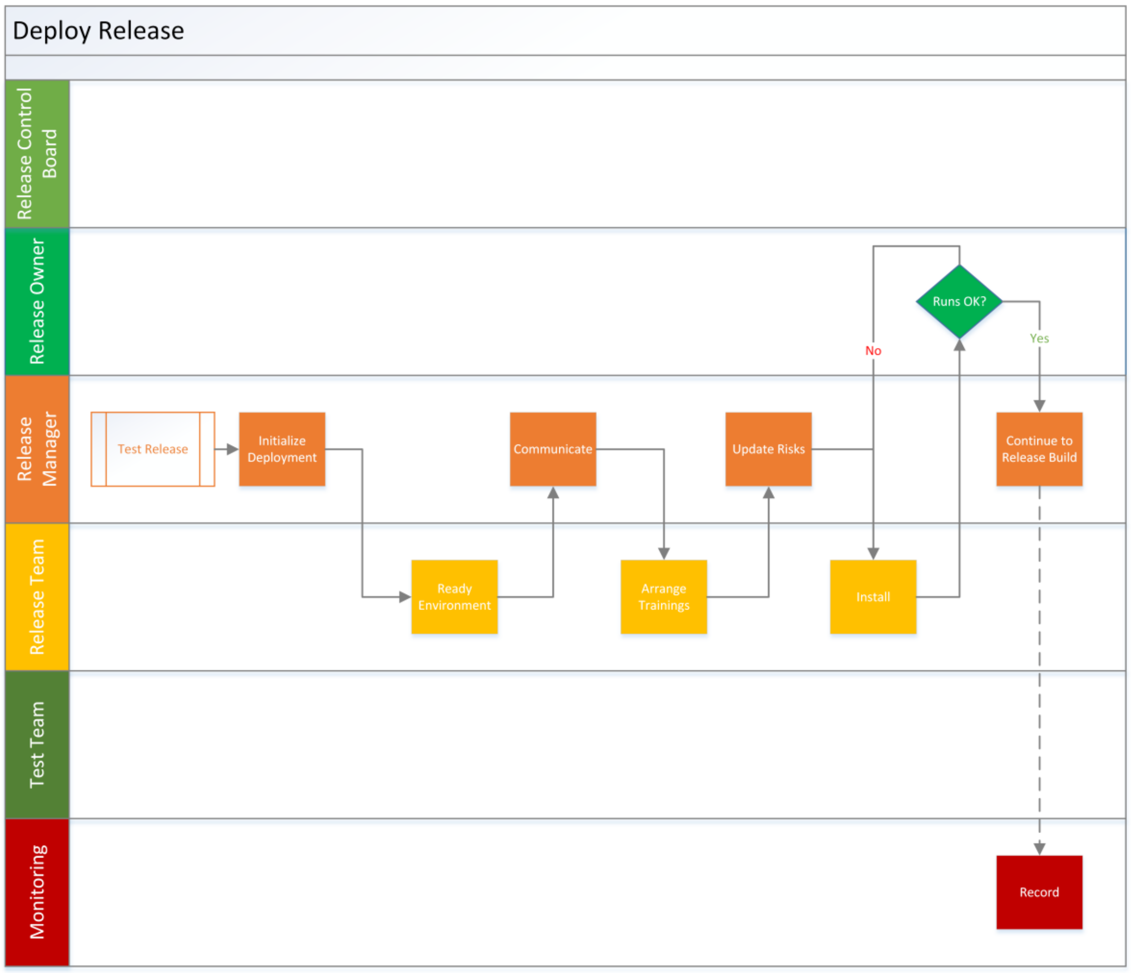
   Acceptance criteria: Test results and the user acceptance testing results.



Picture 21. The release testing phase of the process.

5. Deploy Release (picture 22). The release is deployed to production use following the deployment plan. This stage contains the communication, trainings and required installations.

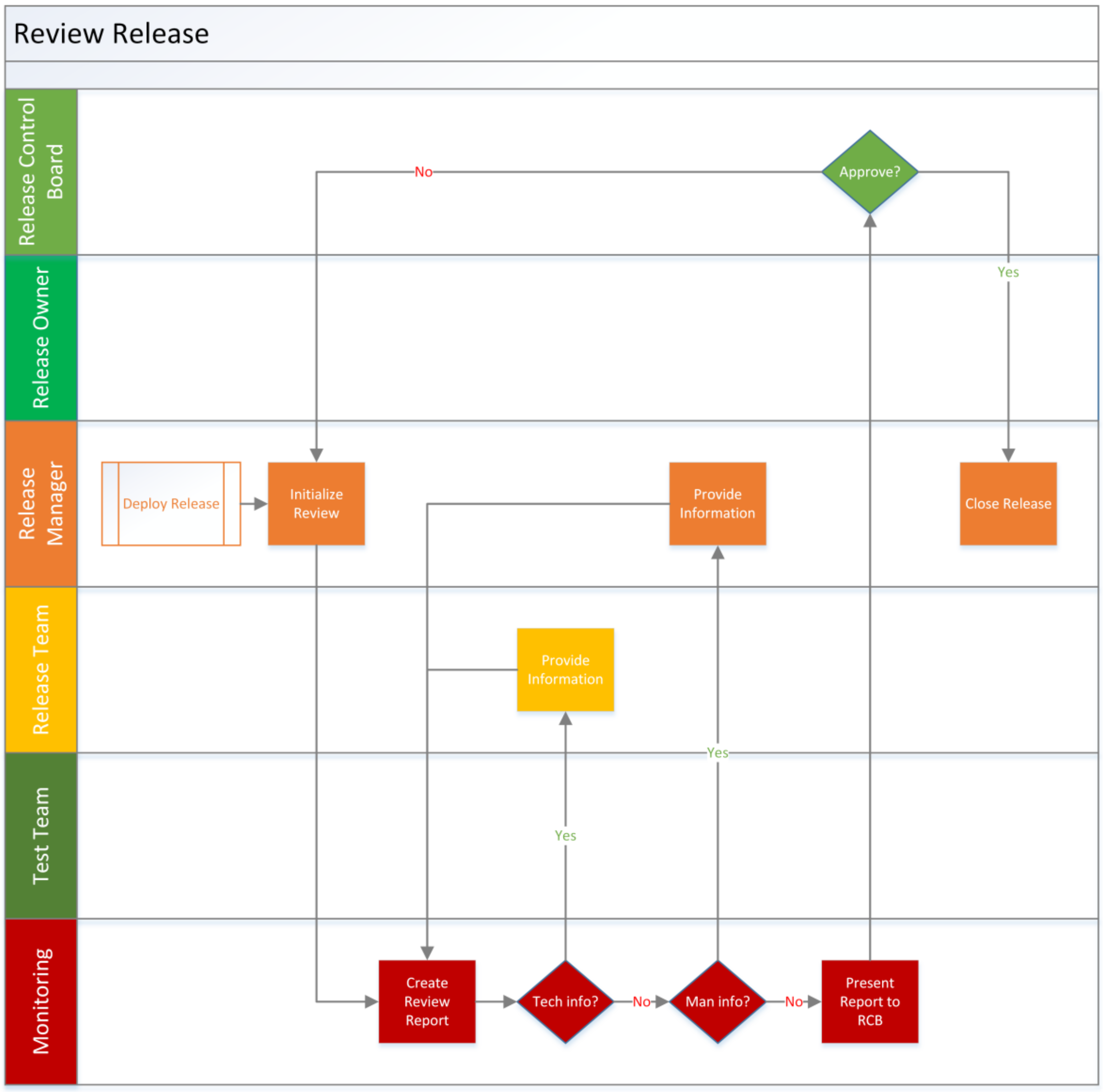
   Acceptance criteria: Successful installation, trainings done, communications done as planned and documentation handed to knowledge management.



Picture 22. The release deployment phase of the process.

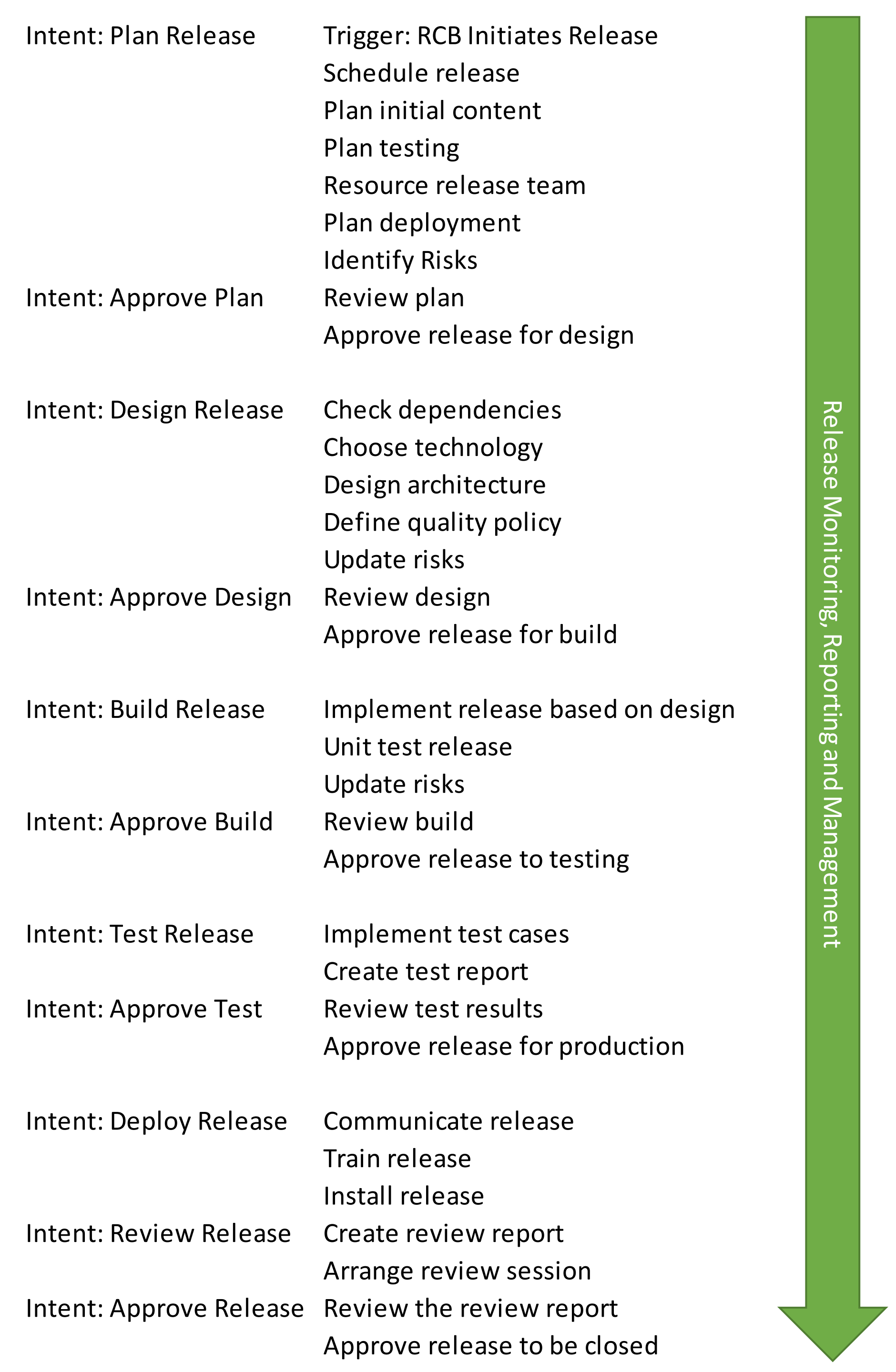
6. Review Release (Picture 23). After the release is ready and stabilized in the production use, the results of the release are reviewed: was the release successful, how it performed, what the level of quality is and what the level of user satisfaction is.

   Acceptance criteria: User satisfaction survey done and the release review report handed to knowledge management.



Picture 23. The release review phase of the process.

7. Release Monitoring, Reporting and Management. The release management process is constantly monitored and the monitoring observations are reported to the release manager. The release manager has a holistic view to the release or releases and is provided tooling to do this effectively.



Intent: Plan Release	Trigger: RCB Initiates Release
	Schedule release
	Plan initial content
	Plan testing
	Resource release team
	Plan deployment
	Identify Risks
Intent: Approve Plan	Review plan
	Approve release for design
Intent: Design Release	Check dependencies
	Choose technology
	Design architecture
	Define quality policy
	Update risks
Intent: Approve Design	Review design
	Approve release for build
Intent: Build Release	Implement release based on design
	Unit test release
	Update risks
Intent: Approve Build	Review build
	Approve release to testing
Intent: Test Release	Implement test cases
	Create test report
Intent: Approve Test	Review test results
	Approve release for production
Intent: Deploy Release	Communicate release
	Train release
	Install release
Intent: Review Release	Create review report
	Arrange review session
Intent: Approve Release	Review the review report
	Approve release to be closed

Picture 24. The process step model of LARMP.

Components 1-6 can be described as process stages or process steps (see picture 24). Component 7 is supporting the whole process by monitoring and reporting it. The management is done with an external view from this component to give more perspective and means to affect the performance positively. The components should be utilized in different types of releases in following way:

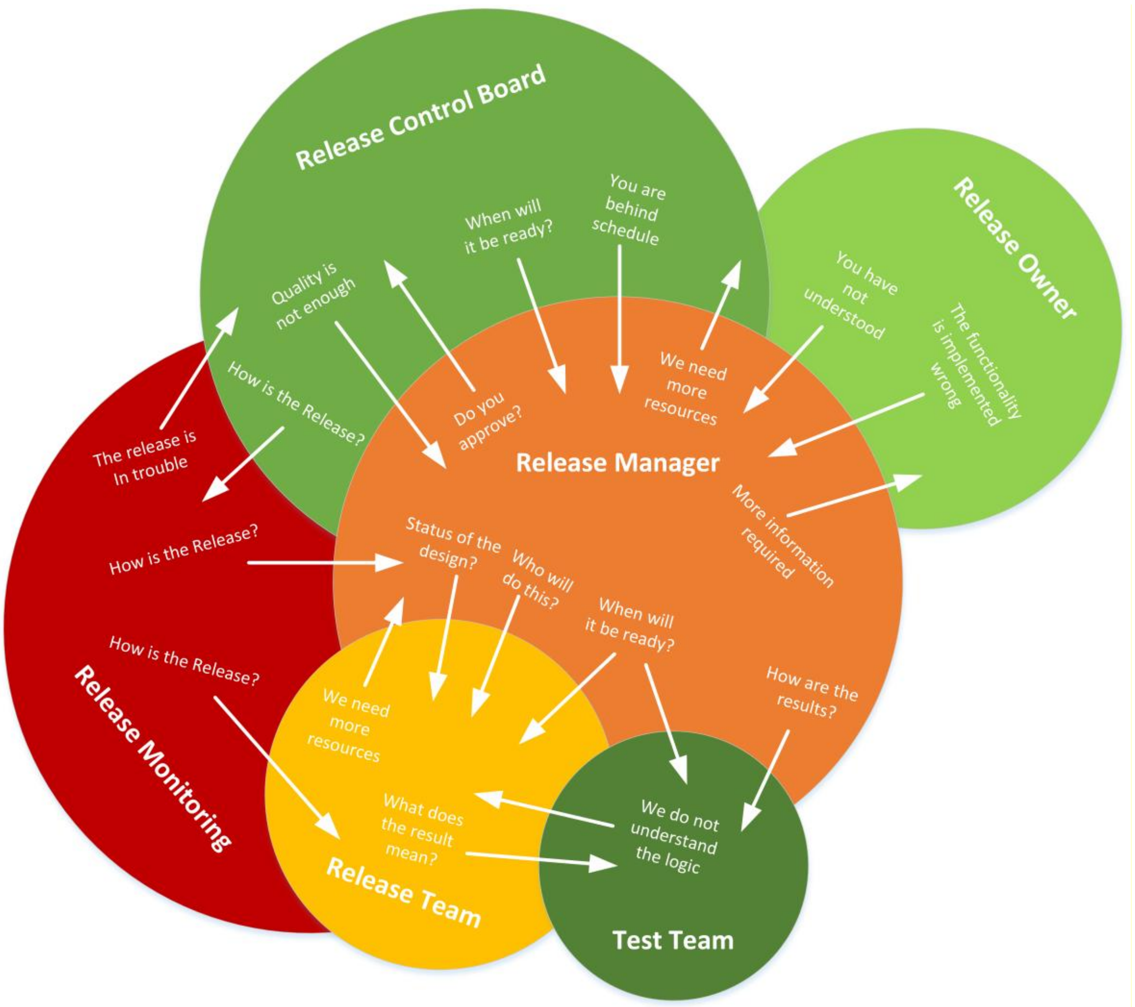
- Emergency release: 2, 3, 5 and 7. There is no time for planning but the process starts from the design and goes straight to building the release. The control points are made quick as the Release Owner can give the approvals. Only unit testing is done. No review is done, but the process is monitored.
- Maintenance release: 1, 2, 3, 5 and 7. As the changes in maintenance release should be minor, no actual testing phase is performed. The review is also dropped as the maintenance release is business as usual and generic. Approvals made by release owner.
- Major release: all of the components. The major release requires the most project like approach and thus all components are applied.

6.3.2. Roles

The cultural modelling technique helps us in the role definition (see picture 25). IT services delivery is affected very largely by cultural factors. As there are usually two or more corporations involved (multivendor environment), there are many cultural factors:

- Different ethnical cultures: IT vendors and also the enterprises might have a global reach and also desire to drive processes globally.
- Different localities: For example the western culture might not vary from country to country, but all of the different locations have a slightly different point of view to work and work processes.

- Different corporate cultures: Different corporations have different values and strategies. These affect the behaviour of every actor in the processes.
- Different business cultures: IT is very different from any other business area. It is almost always in a supporting position and the other business areas have higher status.



Picture 25. The cultural model of LARMP.

Only the corporate culture might be mapped in writing, but that also only partly. These cause clashes and affect the performance of the process. The roles that are defined should take these factors into account.

Four different roles are defined for the process:

- Release owner is the person who owns the configuration items changed in the release. Should be aware of the content of the configuration items and the purpose of the changes targeted to them. The release owner is usually a representative of the business.

  Applicability: All types of releases

- Release manager is the project manager for the release who is responsible for the schedule, quality and financials of the release. Release manager should have a holistic view to the release and also possess the same attributes and competences as a traditional project manager. Depending on the size of the release, it might have several release managers. Or vice versa one release manager can be responsible of several releases.

  Applicability: All types of releases

- Release control board (RCB) is the body, which makes the decisions in the process. The release control board members are the release owner, release manager and any other member seen fit to participate. The release manager presents the cases in the release control board meetings, which are held minimum at the control points.

  Applicability: Major releases

- Release team is a real team or a organization consisting of different operators in the organization. The team could contain a risk manager, change manager, developers and others depending on the release. The more integrated and dedicated the team is the better change for success the release has.

  Applicability: Major releases

- Test team. The test team is defined separately of the Release team as the test team should in every case contain members from the business or customer side. The test team does not do unit testing.

  Applicability: Maintenance to major releases

- Release monitoring is responsible to actively follow a release or releases and report any anomalies in the process to the release manager who can in turn escalate the issues to the release control board.

  Applicability: All types of releases

6.3.3. Input and Deliverables

Every process requires some inputs. Every process should produce some outputs. Without these, there is no need for a process. The outputs from release management or any IT service are usually called deliverables.

The inputs for this process are:

- Requirements. The actual customer needs should be presented as requirements. These cannot be the initial requirements from the customer but the analyzed requirements, which are processes via the requirements management process. Release management should not accept any unpurified requirements as inputs.

- Prioritization. The requirements should be prioritized by the release owner. This way the release roadmap can be defined to deliver on the requirements when they are required to be implemented.

The deliverables of the process are:

- New and changed configuration items. The main purpose of the release management process is to produce the actual release. This release consists of new and changed configuration items. These are delivered to the production system after testing and acceptance by the deployment sub process.

- Documentation of new and changed configuration items. The changed and new configuration items should be documented as described in the release strategy. The minimum documentation should be that the changes are reflected in the configuration management database. At the most extensive form the changes should be documented in the document management system, configuration management database and also knowledge management database.
- Release review report. The release plan is not an actual deliverable of the process but an operational document in the process. The release review report will contain some of the content of the plan and also report of the results. The most important part of the report is the lessons learned part, which is used for improving the process.

6.3.4. Process Control Points

Process control points are defined to provide a mechanism for checking the progress of the process and making appropriate decisions. Without these control points there would not be any sense to define any process. The control points for this release management process are basically the meetings where the RCB convenes to make decisions:

- Approval of the Release Plan: the release plan should be created based on what is defined in the release strategy. The RCB will review the plan against the strategy. The release manager will present the plan to the RCB on the appropriate level that the scope and size of the release demands.
- Approval of Release to Test: The release manager or an appointed member of the release team presents the candidate build of the release to the RCB. The RCB reviews the build against the requirements given to the process.
- Approval of Release to Production: The release manager of the test manager presents the results from the test process and the user

acceptance testing. RCB approves or rejects the release to production use. If the release is rejected, it will be passed back to the build phase with defined correction points.

- Approval of the Release Review Report: The release manager presents the results of the release to the RCB. The release will be reviewed against the plan, requirements, test results, monitoring reports, and the KPIs (presented in the following subchapter). The release manager provides the required improvement actions if any are required. When the review has been approved, the release can be closed. This control point is arranged some defined time after the deployment of the release, for example after two months of active production use.

6.3.5. Key Performance Indicators

Various KPIs could be defined for the Release Management process. These are used in the release review and for the improvement actions.

- Errors in the released software. After the release is deployed to production use it will probably generate some incidents. These are analyzed in the problem management process and are input to the database as known or new errors. These are followed by this KPI.
- Release timeliness. This KPI counts how well the release performed against the defined schedule: was the schedule on time, how much it was on overtime, was it completed too early and what was the performance related to the four different control points.
- Number of rejections from the RCB. The function of the RCB is to guard the quality of the release and reject the release in the control points if it is not ready to be transferred to the next stage. The number of rejections and the reasons for rejections are followed with this KPI.
- User satisfaction after the deployment. The user satisfaction is measured with a survey after some predefined time of production use. Depending

on the scope and estimated user amounts, this time can be from one to six months. The user satisfaction is measured on the actual contents for the release but also for the whole application. Numerical and textual feedback is requested and from there variance and total values are calculated.

All of these KPIs are reflected on the scope and size of the release when used for the continuous process improvement activities.

6.4. Release Plan

The release plan is the living operational document in the release management process. The release master plan defines what the content of each plan should be:

- Overview and purpose of the release: Definition what the release is all about and what are the preconditions for the release. The release strategy is referenced in this section.
- Release organization: This includes the definition of the release team and RCB members. This section also describes the governance model with the release policy.
- Configuration Items in the Release: Listing of the initial set of the configuration items in this release. This should hold the description of the item, CMDB identification and the link to the CMDB entry.
- Schedule: The scheduling should hold at least the dates for the four control points and the deadline for adding new content to the release.
- Risk management plan: This section should include the risk assessment and the mitigation actions. There should be also the back out plan if the risks realize.
- Test plan: The test plan can be a separate document and referenced here or it can be included in the release plan in its entirety. This depends

on the scope and size of the release. The testing plan should contain the test schedule, test cases and plan for resourcing.

- Deployment plan: The deployment plan includes the communications plan, trainings plan and installation plan. The basic purpose in this section is to describe how the release will be transferred successfully into production use.

This content can be tailored for each release. Some of the content can be removed for the smaller releases and maintenance releases. The plan for emergency releases should be kept short.

6.5. Continuous Improvement of Release Management

Each release produces information for continuous improvement of the release management process. This is given in the form of the release review report. The improvement is carried out following the Deming's circle, the best process development practices, ITIL and Six Sigma.

The ITIL Continual Service Improvement (CSI) process covers the continuous service improvement topics. There are two major areas within every organization driving improvement: aspects which are external to the organization such as regulation, legislation, competition, external customer requirements, market pressures and economics; and, secondly, aspects which are internal to the organization such as organizational structures, culture, capacity to accept change, existing and projected staffing levels, unions rules and so on. In some cases these aspects may serve to hinder improvement rather than drive it forward. (OGC, 2007)

Six Sigma is based on statistics and it usually refers to the standard bell-shaped curve for partterns of variations. In statistics, the Greek letter sigma is used to detone one standard deviation. In release management process improvement we follow the KPIs, namely the four defined earlier: errors, timeliness, the rejections by RCB and user satisfaction. In this case Six Sigma refers to the variation on one of these KPIs. The goal of using Six Sigma here is to reduce the deviation from the mean and also improving the mean. As the KPI performance of releases vary based on their scope and size, the KPIs are adjusted based on the scope and size to gain comparable values. (Harmon P., 2007)

Six Sigma refers to the DMAIC process in process improvement (Harmon P., 2007). It stands for

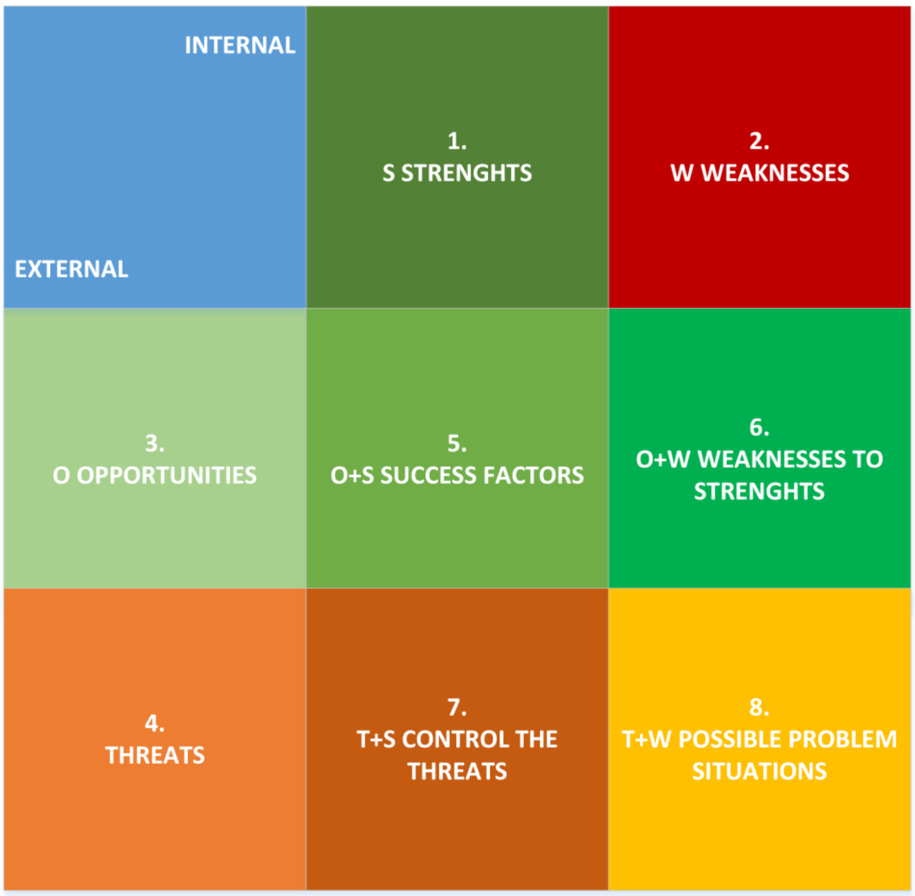
- Define customer requirements for the process or service. The customer is an enterprise and the release management process is built for them. It is good to know what their targets are regarding release management process that is being set up.
- Measure existing performance and compare with customer requirements. If the customer has an existing process, request measurement data on that process. Compare this to the existing release management processes in operation.
- Analyze existing process. If existing, analyze the enterprise process and compare it to the existing release management processes in operation.
- Improve the process design and implement it. Do not take the original enterprise process as template but use the current release management template. Try to fit the enterprise process to the template in the best possible way.
- Control the results and maintain the new performance. See that the gained improvements work and also that the improvement is not lost due to not maintaining it.

All of these steps should be implemented when a new enterprise is integrated to the release management environment. These improvement steps are executed whenever new enterprises are added to the release management environment and whenever major releases are completed.

The SWOT (Strengths, Weaknesses, Opportunities, and Threats) analysis is used for defining the improvement opportunities. We will use the extended SWOT, also called TOWS (Kendrick S., 2011), in this case. It extends the SWOT model in the following way:

- O + S: What are the success factors
- O + W: How the weak points can be turned into strengths
- T + S: How the threats can be controlled with strengths
- T + W: What are the possible problem situations

See the visualization of the extended SWOT process in picture 26.



Picture 26. The extended SWOT model is a perfect tool for process improvement.

6.6. Planning the Implementation into Operation

The environment and the culture define largely how a process can be taken into use. The culture is the most challenging as it can provide great resistance to change. The new release management process would be hard to accept even if it would be done internally. But now we are implementing an externally operated process. Outsourcing work is always seen as a threat by the enterprise organization.

The implementation into use can be aided by different implementation theories applied usually for information systems:

- Diffusion theory (Baskerville R., Pries-Heje J., 1997). The new process will be accepted as information about it is spreading in the organization. If the first experiences are positive, the easier it is to adapt the organization to utilize the new process.
- Problem levels theory (Star S.L., Ruhleder K., 1996). In this theory the problems in implementation are divided into different levels. The problems may be created by the actual process activities or are based on contextual or political factors. This division helps to prioritize and concentrate to fix problems that affect the process implementation most negatively.
- Ciborran care (Ciborra C., 1996). This theory is based on the notion that the process has to be cared on constantly. The process performance has to attain the level it has reached. The performance of the process has to be made public so it has a good reputation and it can be accepted more easily.
- Orlikowski's duality model of technology (Orlikowski W.J., 1992). Orlikowski presents that people are changing technology but are also using technology. The main aspect in Orlikowski's theory is that if you present some means for the process utilizer to affect the process, the process becomes more meaningful to the utilizer.

- Actor-Network Theory (Law J., 2000). The ANT theory could be used to reveal the people who are pro and who are anti-release management. It also proposes black boxing the process so that the details do not interfere adopting the process.

Based on these theories, the implementation plan should take note on these topics:

- Communication quality and communication timeliness
- Identify the groups that are supportive and that are against
- Proactive problem detection and division of problems to the ones caused by the process and to the ones originating elsewhere
- Continuing process support
- Involve the process stakeholders
- Use process blackboxing where possible

The physical environment in this case means the information systems, which are the target and tools for the release management process. It can be challenging to implement the process to different technical environments if it is highly reliant on some distinct solutions. But we have not described in detail any distinct software for this process so it should be implementable to any suitable software available. The worst case scenario is that the release management is implemented in a spreadsheet, which is a passable option but grows harder by the size of the release.

The release organization will be virtual. There is no requirement of physical presence, but that can be arranged if necessary. The work is done mainly online and aided by information systems, be that even a spreadsheet.

For implementing the process for different situations we could consider using the BPTrends enterprise methodology to first map out the enterprise (Harmon P., 2007). This begins with the creation of series of organization diagrams that define the business and its key relationships and gradually refine everyone's understanding of the organization and its stakeholders. The value chains of the organization are defined. The goals of each value chain and the relationship between core processes and managerial and support processes are also specified. Thus, a specific business process architecture is developed for each individual value chain. This way the understanding of the enterprise grows and the release management process can be adapted more easily to satisfy the needs of the target enterprise.

6.7. Benefits of Outsourcing Release Management

As all IT, also release management can be outsourced. In an outsourcing relationship, one party (the client) asks another party (the vendor) to perform a given task. The outsourcing vendor then assumes the responsibility of completing that task according to the client's guidelines and expectations. (St. Amant K., 2010)

Commonly outsourcing includes some level of offshoring, which aims to cut cost by delivering work from low cost locations. If release management and its interfaces are defined clearly, it is easy to outsource.

Release management is often the last application management process to be outsourced. Enterprises are undoubtedly keen to keep the release control to themselves. The introduced release management process does not take this away. The control of the process is in the hands of the enterprise, but they also have a pool of external experts to support them.

I see following benefits for outsourcing release management:

- Corporate IT can concentrate their efforts to creating new value to business instead of micromanaging the process. The corporate IT uses much time to control their processes and solve problems related to them. Much of this work can be outsourced. Then the corporate IT can use their time and resources to use their competence related to the core business to elevate the business processes.

- The level of control can be set based on the scope and size of the release. When seen fit, the process can even be almost a black box. Then the corporate IT attends only to the decision making in the RCB.
- The release management process is aligned to industry standards making it more effective. The vendors are using industry standards like ITIL and can also apply experiences from other release management activities to this process.
- The releases are implemented by appropriately sized pool of competent resources, which can be resized when necessary. Vendor resources are experts in the area of IT and are trained in release management and related processes. This also removes some training needs from the corporate IT.
- The releases are managed in appropriate manner by experienced release management professionals making the success rate higher. The release management is one of the many parallel release management processes. This provides synergies and also opportunities to increase performance and reduce cost.
- There is more time and resources to improve the process. As there is a dedicated vendor managed release process in use it can also be improved more effectively. This can contractually even be added as a responsibility of the vendor.
- The release management risks can be shared with the vendor. There are many risks in release management. These risks can be even contractually be shared by or moved completely to the vendor. This provides more room for the corporate IT to concentrate on the essentials.
- Flexible cost models can be utilized transferring the responsibility out from own organization. The release management process can be set so that there is for example an unit based costing or a fixed cost per release.

- The process can be sanctioned based on the KPIs. The KPIs that are measured from the process can be a basis for sanctioning. It can be defined separately for each release or to all releases. This adds more pressure to the vendor but removes some pressure and budget risk from the corporate IT.

6.8. Tooling Release Management

The release management process can be hardened by using dedicated or shared tooling. The tooling should be defined and prepared depending on:

- The scope of release management. The first question should be that does the release management process scope require a tool or could the process be handled without one?
- The available tools. There are multiple tools already available in the enterprise. It should be analyzed, if the tools could support the release management process.
- The dedication of the enterprise towards release management. There is no sense of investing on new tools or tooling projects if the process is not utilized.

As release management is not a separate and isolated process, it should be linked to other processes in the tools. Many modern IT service management tools provide the release management functionality out of the box. For example the following popular tools come with release management enabled and linked to other relevant processes like change management (Mann S., 2013):

- BMC Remedy OnDemand
- CA Service Desk Manager
- IBM SmartCloud Control Desk
- ServiceNow

These tools support tailoring as they are cloud based solutions. Based on this they are probable candidates for tools to run the LARMP process. There is also the possibility to build the tool for LARMP as a custom build software. The tool can then be a simple web based solution or even a common spreadsheet solution. But when the scope of release management is larger, an out of the box tool is preferable. This way all of the time used in release management would not be used investigating the problems in the tool.

The social media tools are also something to be considered with release management. When release management is not yet implemented and a social media enterprise tool is available, the options integrating or running the process in that tool should be tested. Advise for the approach in such case would be worth of another thesis altogether. But if that approach is taken, the enterprise is demonstrating a will to be a true example of an enterprise 2.0 (Taborda L.J., 2011).

Choosing the correct tool can bring even more visibility to the process. Release management is then easier to monitor and the possibility to outsource the process is higher. I would personally recommend selecting the tooling as the first step when defining the release management process.

7. Conclusions

Managing enterprise applications has challenges, which stem more from the environment than from the singular applications themselves. Enterprise IT landscapes can be very complex and usually they are not thoroughly documented. There is a really valid reason to define a structured way to approach this challenge. One good tool for this is release management. By using it is possible to gain a certain level of holistic control of an enterprise application landscape.

Application management has a big role in maintaining the enterprise IT architecture. It is supporting the current solutions and also providing power to enhance them. AM has to be given more emphasis so that it can be more effective to keep the systems in good performance and usable. This way the IT resources can be freed to develop new solutions to push the business even further.

The release management process introduced in this thesis is developed to be light and adaptable. Hence the name Light and Adaptable Release Management Process (LARMP). It is not tied to a singular type of technology environment. It is not tied to any singular industry standards either. It is designed to be used wherever there is a need for release management. It can be taken into use with minimum effort. Despite this it can provide maximum benefits of visibility, predictability, risk mitigation efficiency and even financial benefits.

As for ITIL, the starting point for release management in LARMP is the release strategy. It has to be defined for the whole release environment and also for each release separately. The process itself is not complex and it has seven sequential components which are defined with lean principles in mind. The cultural environment is defined to be light as well; there are minimum set of roles with clear responsibilities.

The quality of the process deliverables is maintained with defined control points. It is also good to note that the process is improving: it is vital that mistakes are not forgotten, but are used as stepping stones for improvement.

Benefits of outsourcing release management are many. The main benefit is that the release management is controlled by a dedicated organization, which does not take the ultimate control away. The release management process acts as a quality gate and risk mitigation device. The risk can be divided with the service provider, which also gives the provider more motivation to make the process really work.

The initial intention of this work was to define a release management process. Due the course of the work and related research, it came very evident that release management has an increasingly important function in enterprise IT and also business continuity. It is not the only process in the field but it can be used to introduce more control to application management.

Release management can really be the backbone for enterprise application management. It is a safeguard between the erratic business needs and the working application. It makes sure that the risk of negatively affecting the operation of the application is mitigated to the full. It builds more solid ground for the other application management processes to work on. For example with fully functioning release management it is possible to decrease the amount of incidents.

The enterprise IT is growing more fragmented again in the wake of new trends like the Bring Your Own Device (BYOD) thinking. Corporate IT is losing control of the overall IT environment and is suppressed to a supportive status. Business is eager to make the decisions regarding IT. The presented release management model is one option of many ways to have some IT control at the same time as giving the control to business. Defining a release management process is not enough alone: it has to be implemented with belief that it will present its benefits in the long run.

References

Information Resources Management Association (IRMA)
Enterprise Information Systems: Concepts, Methodologies, Tools and Applications, Volume 3, IGI Global, 2011

Egelhoff, W.G.
Information processing theory and the multinational enterprise, Journal of International Business Studies, Vol. 22, Iss. 3, USA, 1991

Taborda L.J.
Enterprise Release Management: Agile Delivery of a Strategic Change Portfolio, Artech House, USA, 2011

Fyrvik T., Uthaug E.S.
Process Modelling as Basis for Development and Integration of New Information Systems, Martinek Solutions, Norway, 2005

Grinyer P.H., Al-BazzazS., Yasai-Ardekani M.
Towards *a* contingency theory of corporate planning: Findings in 48 UK companies, Strategic Management Journal, 7(3), 3–28, UK, 1986

Columbus L.
2013 ERP Market Share Update: SAP Solidifies Market Leadership, http://www.forbes.com/sites/louiscolumbus/2013/05/12/2013-erp-market-share-update-sap-solidifies-market-leadership/, Forbes, 2013, read 16.11.2013

Office of Government Commerce
ITIL: Service Design, OGC, United Kingdom, 2007
ITIL: Service Operation, OGC, United Kingdom, 2007
ITIL: Continual Service Improvement, OGC, United Kingdom, 2007
ITIL: Service Transition, OGC, United Kingdom, 2007

Myatt M.
Best Practices - Aren't, http://www.forbes.com/sites/mikemyatt/2012/08/15/best-practices-arent/, Forbes, 2012, read 16.11.2013

ASL Foundation
ASL – A Framework for Application Management, Van Haren Publishing, United Kingdom, 2004

Du Preez D.
Government sells 51% stake in ITIL and PRINCE2 to Capita, http://www.computerworlduk.com/news/public-sector/3444373/government-sells-51-stake-in-itil-and-prince2-to-capita/, ComputerWorld.uk, 2013, read 16.11.2013

Meijer M., Smalley M., Taylor S.
ITIL V3 and ASL - Sound Guidance for Application Management and Application Development, TSO, 2008

van Bon J., Pieper M., van der Veen A.
Introduction to ITIL, Van Haren Publishing, United Kingdom, 2005

Howard D.
IT Release Management: A Hands-on Guide, Auerbach Publications, USA, 2010

Sturm R., Morris W., Jander M.
Foundations of Service Level Management, Sams, USA, 2000

Arveson P.
The Deming Cycle,
http://balancedscorecard.org/TheDemingCycle/tabid/112/Default.aspx,
Balanced Scorecard Institute, USA,1998, read 16.11.2013

Klosterboer L.
Implementing ITIL Change and Release Management, IBM Press, USA, 2009

Keyes J.
Software Configuration Management, Auerbach Publications, USA, 2004

Jennex M.E.
Knowledge Management in Modern Organizations, IGI Global, USA, 2007

Davenport T.H., Prusak L.
Working Knowledge: How Organizations Manage What They Know, Harvard Business School Press, USA, 1998

Beyer H., Holtzblatt K.
Contextual Design, ACM Press, USA, 1996

Harmon P.
Business Process Change: A Guide for Business Managers and BPM and Six Sigma Professionals, Second Edition, Morgan Kaufmann Publishers, USA, 2007

Kendrick S.
The TOWS Matrix: Putting a SWOT Analysis into Action
http://www.volunteerhub.com/blog/the-tows-matrix-putting-a-swot-analysis-into-action/, VolunteerHub, USA, 2011

Jäntti M. Lahtela A.
Challenges and Problems in Release Management Process: A Case Study, IEEE, Finland, 2011

Jäntti M., Sihvonen H-M.
Improving Release and Patch Management Processes: An Empirical Case Study on Process Challenges, Fifth International Conference on Software Engineering Advances, Finland, 2010

Baskerville R., Pries-Heje J.
IT diffusion and innovation models: The conceptual domains, Proc. IFIP WG8.6, Chapman & Hall, UK, 1997

Ciborra C.
Groupware & teamwork: Invisible aid or technical hindrance, Wiley, UK, 1996

Star S.L., Ruhleder K.
Steps toward an ecology of infrastructure: design and access for large information spaces, Information Systems Research 7, USA, 1996

Orlikowski W.J.
The Duality of Technology: Rethinking the Concept of Technology in Organizations, Organization Science, USA, 1992

Law J.
Actor Network Resource: Thematic List, http://www.lancaster.ac.uk/fass/centres/css/ant/ant.htm, Lancaster University, UK, 2000, read 17.11.2013

St. Amant K.
Understanding IT Outsourcing - A Perspective for Managers and Decision Makers, IGI Global, USA, 2010

Mann S.
Market Overview: SaaS IT Service Management Tools, Forrester Research, USA, 2013