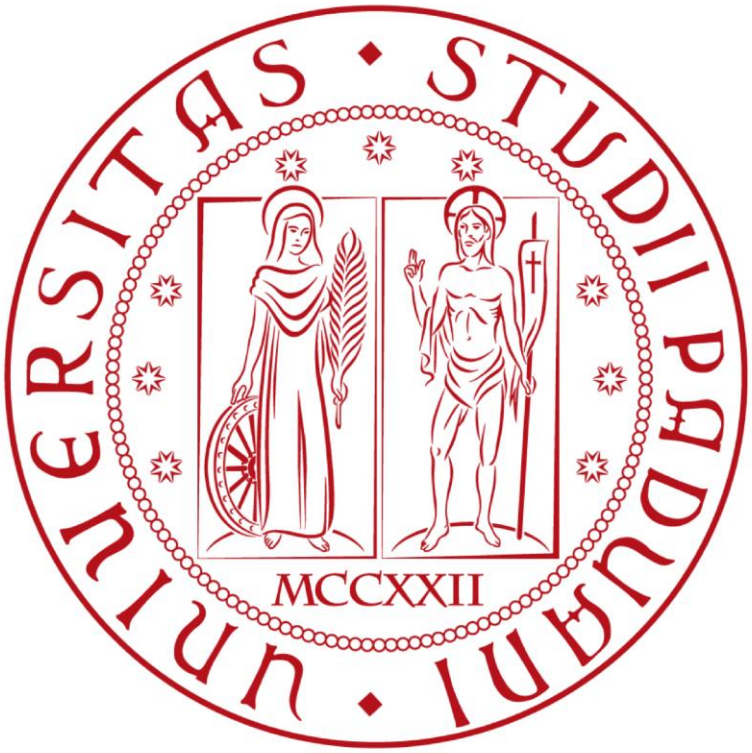

IMPLEMENTING SAAS CRM NDB (NEW DEVELOPMENT BANK)

Presented By Adnan khan

Matricola Number : 2041575

NEW DEVELOPMENT BANK

ITSM NDB SaaS Release Plan

Plan Details			
Description:	The new version is intended to implement Salesforce SaaS CRM module for the New Developement Bankcin response to their RFP.		
Plan Owner:	Adnan Khan (PM)	**Role:**	Project Manager
Submitted On:	30 June 2022	**Submitted To:**	NDB Information Technology Division
Release Type:	Major Release	**Release number:**	1.0
Status:	Pending Approval	**Risk Level:**	Medium
Owner of release (PM):	Adnan Khan	**Start Date:**	1st July 2022
Duration:	8 weeks	**Planned Release Date:**	13 September 2022

Changes Included in the Release	1. The new SaaS version will include many more automatic workflow processes and allow us to focus on automating processes rather than putting them together. 2. The CRM modules will be built-in and a lot less development will be required. 3. The new authorization capabilities will allow module IT department grant/alter permissions easily without any customization.

Who Will be affected by the Release	1. All the CRM dashboards users. 2. CRM developers. 3. Module owners managers who grant authorizations.

Potential Risks	1. All the dvelopments in the previous version's code may not be compatible with the new version and require updates. Mitigation: Purchase as automated system which will update the code 2. Authorizations in the new version won't be the sense as in the old one. Mitigation: Map the current authorizations per user and role.In case we will need to reduce them on the new version.

ITSM NDB SaaS Release Plan

Approval Chain for Change Requests (CR)	All new CR's must be sent to R&D Release PMO.The chain of approval will then be as follows:- 1. Release Manager 2. CIO 3.VP R&D 4. CAB Once approved by the CAB, the CR can be implemented.

TABLE OF CONTENTS

1. INTRODUCTION:

During the last decade we have all experienced how goods and services have been digitized and virtualized. Not only are these tangible things transforming, software – the driving power of many of these goods and services – is also impacted by this trend. The question arises as to how a class of products that is natively digital has been transformed in the way it is delivered and consumed. Think about the tablets (iPad) and the way of using software. People no longer need to know how to install and use a software to run certain tasks. This can all be solved now with the tip of a finger on an icon. How is this possible? It is all due to the Internet, that connects people to people, businesses to businesses, and people to businesses. Today, this allows us to do things that would not have been possible 10–15 years ago. Therefore, it was only a matter of time before the next logical step took place. The already digital product software became even more digital by giving up its last physical resort – the (CD/DVD) – and can now be provided via the Internet. Since the beginning of the new millennium software solutions for e.g., accounting, collaboration, CRM have been provided via the Internet i.e., no longer must be installed on a computer, can run via a browser only and are offered in a pay-per-use mode.

2. LIST OF ACRONYMS:

CI	Configuration Item
CMS	Configuration Management System
DSRM	Design Science
IS	Information Systems
IT	Information Technology
ITIL	Information Technology Infrastructure Library
ITSM	IT Service Management
KEDB	Known Error Database
KPI	Key Performance Indicator
RFC	Request For Change
SLA	Service Level Agreement

3. ITIL IN SAAS CRM:

Despite ITIL not being support out-of-the-box, the fact that SaaS CRM has a workflow management system makes it easy to add several processes such as Incident Management, Problem Management and Change Management by using SaaS built in tools. Other processes such as Release and Deployment Management are partially supported due to SaaS integration with version control systems and deployment capabilities provided by the add-ons e.g., Bamboo.

4. NEW REQUIREMENTS – UPHEAVAL IN THE BANKING INDUSTRY:

SaaS (Software as a Service) is a software delivery model in which software and the associated data are hosted centrally and typically accessed via the Internet. CRM is – and continues to be – the largest market for SaaS, and many enterprises already rely on the flexibility and stability SaaS technology provides. With SaaS all hardware and software components are installed, tested, and maintained by third party providers: there are no complex, labor-intensive implementations and ongoing support which utilize internal IT resources. Traditional 'On Premise' systems – which are still in place in many companies – had their peak in the late 90s, and the companies used the systems' deep customization capabilities to tailor the features, interfaces, and other characteristics of their solution to best support their specific needs. Long implementation and integration projects with many internal and external IT resources were set up, resulting in a highly customized solution that could only be changed or enhanced with significant human and financial effort. Staff resources had to be dedicated to ongoing system monitoring, as well as routine services, upgrades, and enhancements. The systems which were implemented a decade ago are not flexible enough to handle modern-day demands and to react quickly to changing market requirements. One of the biggest shifts in response to changing market requirements will be the way companies sell their products and the way the customer, the professional as well as the consumer, is involved and approached within this process. Multi-Channel-Management capabilities will become more important in order to intensify the interaction with the customer through an efficient coordination of all contact channels across the entire organization. Customer focusing processes will not be left exclusively to Sales and Marketing. Instead, other organizational entities such as Banks, Market Access and service units will be involved in driving customer centricity. This means that the company must consider multiple internal groups with specific requirements. Consequently, established processes will have to be revised and adopted or even replaced by new process models. This new approach demands novel technology, which is flexible, fast, specialized, and much less expensive to deploy and maintain – a technology that finally helps the organization to cater to their customers' needs. Implementing a SaaS-based CRM system brings many benefits to the enterprise, from high scalability to multi-tenancy and greater flexibility, which allow the

company to react quickly to market demands while facing a significant decrease of resources in the internal support organization. When applying a 'rapid development' approach SaaS-based systems are much faster and less expensive to deploy and maintain.

Moving from 'On Premise' to 'On Demand' by introducing a SaaS platform is not only a technological switch, but also an opportunity to drive the enterprise towards more harmonization and standardization. Salesforce.com, which coined the term 'The End of Software' to differentiate the new SaaS approach from the traditional 'On Premise' application, has contributed significantly to making SaaS popular. However, before starting developing and implementing such a system, several aspects require consideration to promote a seamless deployment to the organization and make the initiative a long-lasting success story.

In the following some important aspects which should be considered prior to implementation work will be discussed. To allow better illustration the further description is based exemplarily on an implementation of Salesforce.com (SFDC) for a new development bank NDB. This needs to be completed in 2022 and can be summarized as follows

- International roll out to 5 countries
- Number of users by country organization ranging from 500–3000
- Centralized approach, governance, process scope, system development and support
- are defined and managed centrally
- Harmonized processes, utilizing 70 per cent standard processing for core template and adding 30 per cent of country specific requirements.

5. GENERAL ASPECTS WHICH REQUIRE CONSIDERATION:

SaaS is a software application delivery model where the software vendor develops a web native software application and hosts and operates the application ready for use over the Internet. However, to make an 'out of the box' platform useable to an enterprise, various aspects must be considered while also reflecting on the requirements for adoption. The initial and also most important task – especially for a multi-country roll out into an international environment – is the proper definition of the organizational structure, which will later determine the scalability and accessibility of the system and can only be reversed with difficulty. Even though a 'rapid development approach' could be applicable from a mere implementation point of view, the setup of a conceptual phase prior to the implementation is highly recommended.

During the conceptual phase the following aspects have been worked out thoroughly:

5.1 Governance

- Concept for Governance Committee
- Allocation of responsibility in Business & IT

5.2 System Landscape

- Evaluated number of physical orgs required
- Considered local, regional, global definition of organizations
- Organizations are determined: geographically, functionally, data wise

5.3 SFDC Edition

(e.g., Professional, Enterprise, Unlimited): functionality, data storage, customization limits, development environments (sandboxes) depend on the Edition to be purchased.

5.4 Data protection requirements

- Ascertained if the Safe Harbor Agreement (data is stored in an environment outside the premises) is sufficient for the organization
- Evaluated if local data protection and compliance rules apply.

5.5 Understood and evaluated basic structural elements of the system, especially for cross country implementations or the deployment of a complex organizational structure.

- Roles & Profiles
- Record types
- Page layouts
- Field level security

5.6 Evaluated and understood limitations within the system (number of fields, objects, snapshots)

5.7 Standard vs. Customizing: define threshold for use of SFDC Standard Functionality vs Customizing vs. Coding (Apex/Visual Force)

- Evaluated and defined level of standardization across countries
- How much adoption of functionality have accepted for a single country?
- How much 'standard' a single country accepted?

5.8 Reporting

- Evaluated the level of standard KPIs and reports
- Evaluated use of SFDC reporting engine vs. separate reporting tool

5.9 Use of mobile devices

The following benefits could be leveraged with an accelerated project approach:

- No additional cost for hardware.
- No assessments needed with regards to sizing or infrastructural questions.
- Hosted centrally, new releases are put in place without requiring users to install software.
- Single configuration, making development testing faster.
- Integration with other systems through API.

6. HOW TO MANAGE 'MULTIPLE-COUNTRY' APPROACH:

Due to the increasing harmonization of organization and processes across countries, many companies consider an international roll out of a CRM platform. Applying this approach, the desired level of harmonization and standardization, relating to content and technology, must be determined. The following measures are the first steps taken towards cross-country harmonization:

- **Definition of Strategy and Processes:** define 'core' under consideration for local characteristics.
- **Scope definition:** identify high-level requirements and prioritize, with involvement of key stakeholders and/or Steering Committee.
- **Set standards** for all BRICS countries, in terms of organization, processes, KPIs.

Finding a balance between global goals and the fulfillment of local needs is the most challenging part of the initiative. However, focusing on adjustment to an appropriate level will be a prerequisite for making the entire project a success. For a multiple country roll out a pilot approach is the most appropriate for validating the success of the conceptual pre-work. The pilot should – on a high level – reflect the process scope and represent the desired organization.

- Implementation of one or two reference countries as proof of concept
- One smaller country with a high level of standard functionality and limited interface requirements
- One large, complex country with individualized requirements and interfaces to local systems
- Use of standardized templates and toolkits to request functional details from the selected countries for preparation of the prototype

- Derivation of the set up for a global implementation project based on this experience, carefully considering the sequence of countries to be rolled out with a multi-country approach as defined above a global governance concept can be applied without disregarding local flexibility.

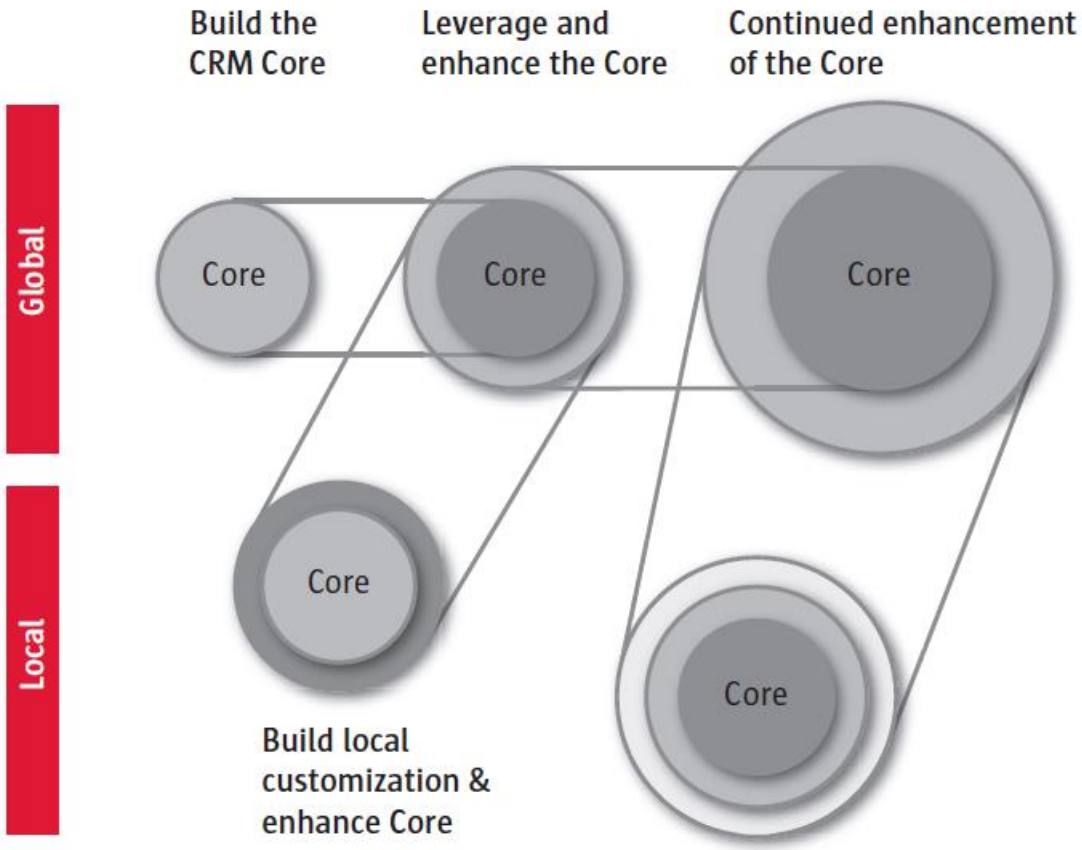

Figure: 1 Developing a global core with local input

With an appropriate release concept, the country specific requirements be integrated to fully enhance the core application.

A multi-country SaaS roll out for a global organization will be executed as shown in the following Figure 1.1.

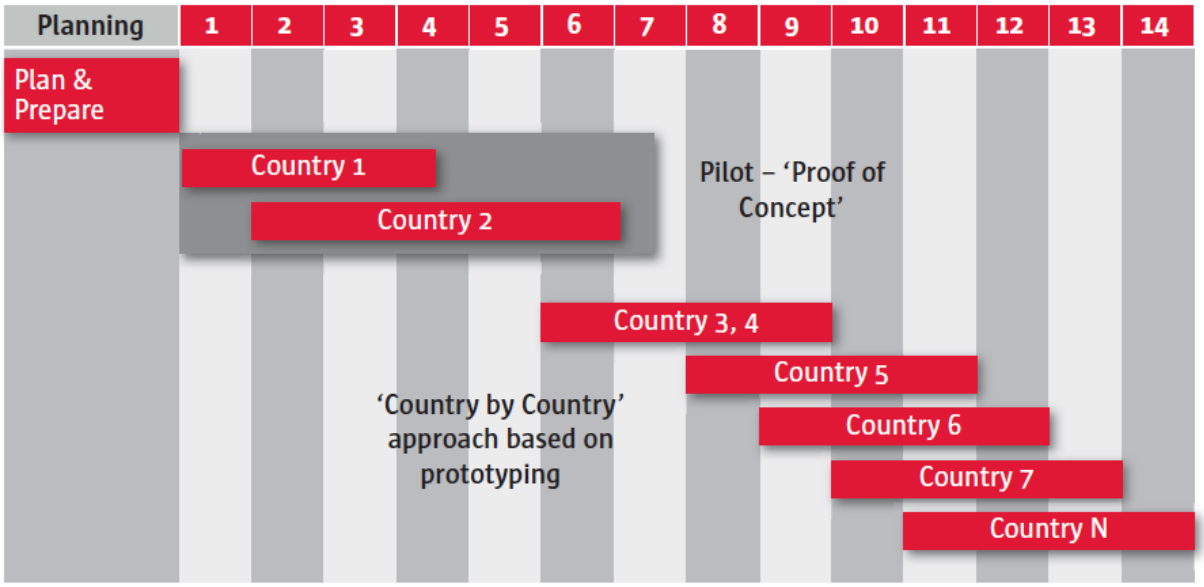

Figure: 1.1 Parallelization of country roll outs

For each country considered for the global roll-out plan the same pattern will be applied (if countries are of similar character some countries could be clustered, e.g. Nordics region):

- Gathered basic business specifics from each country by using a pre-defined toolkit.
- Developed a high-level country-specific prototype when setting up the organization (Using record types, profiles, roles).
- With each country or country group run a detailed requirements verification session.
- Enhanced the prototype taking country-specific requirements into account (changes to the prototype can often be implemented 'on the fly' during a requirements' verification session).
- Conducted multiple reviews of system prototype and discuss options for the implementation.
- Testing, training, data migration.

A generic implementation plan for a single country will be as below in Figure 1.2.

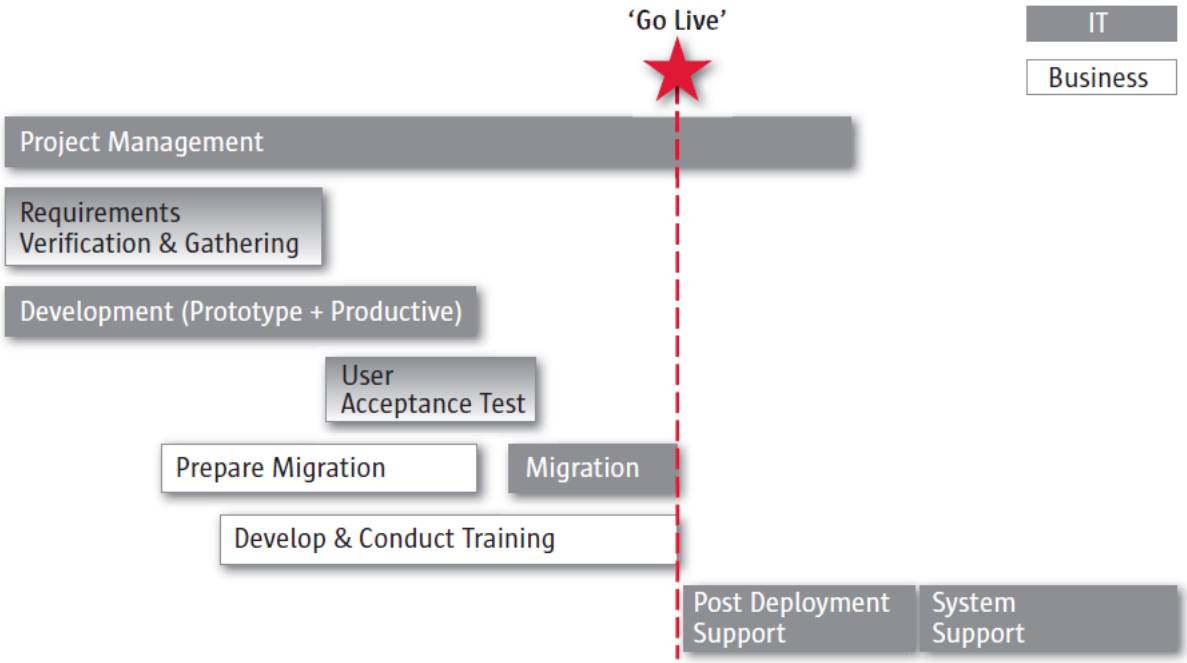

Figure: 1.2 Generic Project Plan for one implementation by country or country cluster

7. GOVERNANCE:

To gain the most benefit from a multi-country implementation scenario a centralized Decision Committee that assesses and ranks input from the countries for global use will be set up. The Steering Committee will act as the superior instance giving direction on strategies, scope and providing resolution for serious issues. The steering organization will be set up as shown in the following Figure 1.3.

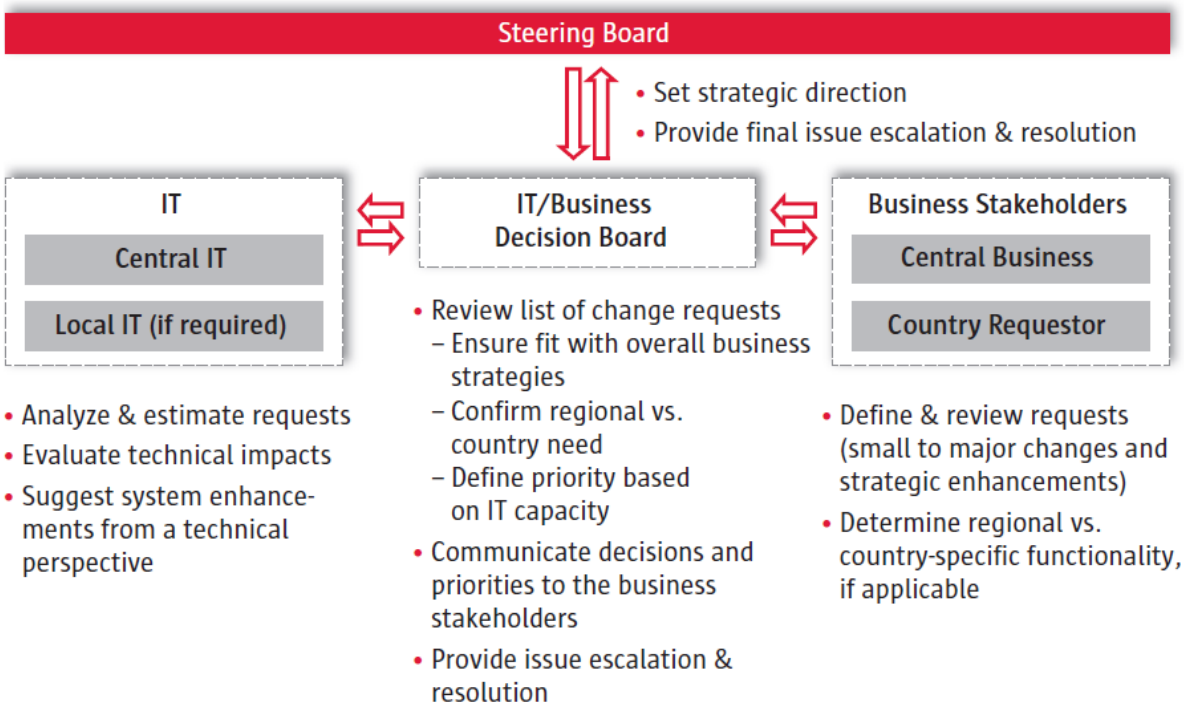

Figure: 1.3 Setting up the steering organization

We supported the decision-making process by making mandatory that an appropriate communication process between all involved parties is established. This includes carefully leveling out the involvement of Business and IT and the build in of feedback loops with the user community.

After the roll out, when everybody is getting used to the new system, new ideas will arise, and the country organizations will submit support and change requests to IT. To process all requests on time, an appropriate support concept needs to be in place, outlining the workflows, responsibilities, and Service Level Agreements (SLA).

The following structure for a support organization has proven successful below in Figure 1.4.

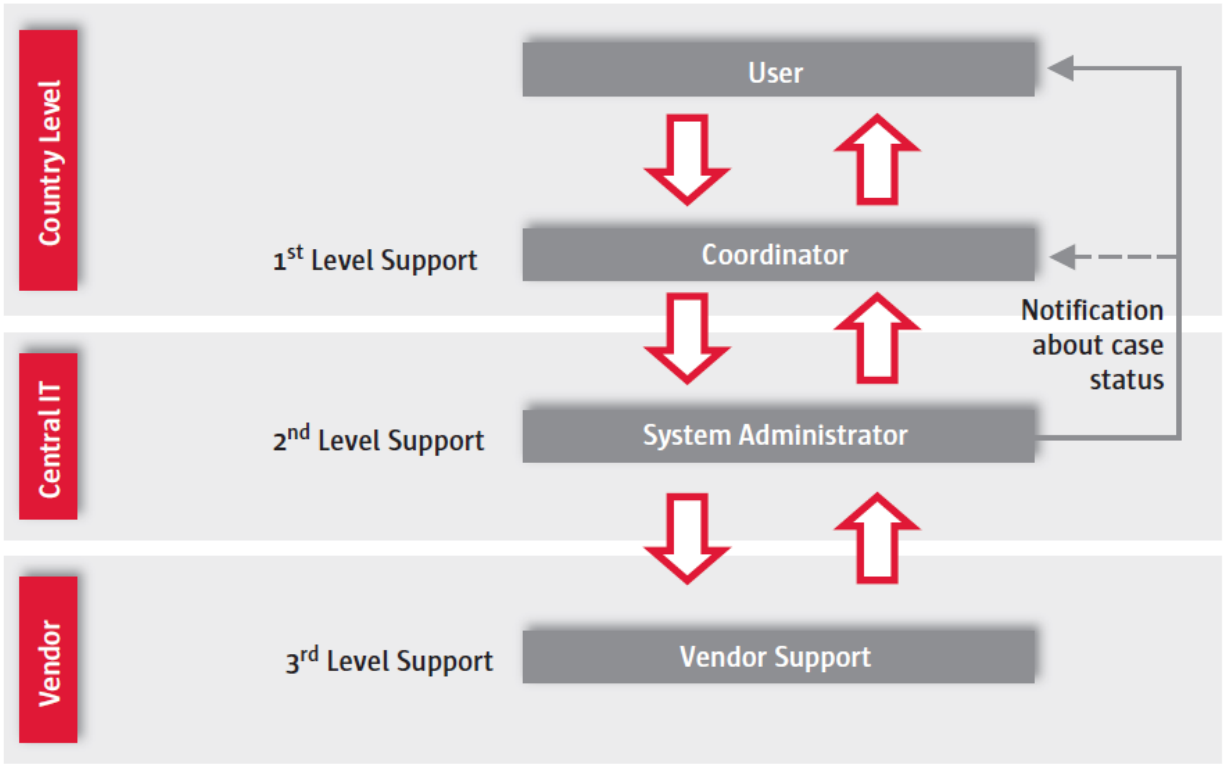

Figure: 1.4 Generic model for support organization

For a multi-country approach a centralized support organization is the most obvious solution. However, a coordinator within each country will be established to monitor and control change or enhancement requests addressed by the country's business organization.

- **Country Level:** Business Coordinator per country as first point of contact for the end users and single point of contact for IT. The Business Coordinator will manage the following tasks: system training (initially and with new employees), assessment of requests from business and creation of support cases, suggestions for further development of the system, testing, active participation in go-live, management reporting.

Additionally, the Business Coordinator will already play a central role for the implementation project and understand the country specific requirements.

- **Central IT:** Technical support resource to assess and implement support cases.
- process operational tasks: user management, passwords, changes to territory structure, new products, small changes/change requests etc.
- assess small change requests – implement directly in the production environment.

- assess changes/enhancements with high effort: use of sandbox environment for development and test, then migrate to production.
- **Vendor:** Third level support for issues or change requests that cannot be resolved without the vendor's input.

8. DEMONSTRATION:

My proposal is to implement the processes that provide quick-wins, that is, provide visible improvements and benefits in the short term for NDB.

In this section we will begin by describing the implementation of the selected processes in SaaS CRM, beginning by describing how we will create and configure each project and moving on to a detailed description of how we will tackle each peculiarity of the processes. It is noteworthy to mention that Knowledge Management, Service Level Management and Service Desk don't have a project as knowledge resides exclusively in Service Level Management and Service Desk are implemented inside the other processes' projects. At the end of the present section, we will present a small description of the used add-ons and **pricing** for an organization with users per month usage as to provide an estimation of the cost of acquiring the needed software for our proposed solution. To accomplish this goal, we decided to encapsulate each of the five processes, and the related Service Desk function, into a separate project. We chose this approach in order to isolate each process thus reducing the complexity of each process while retaining the ability to share components between them when appropriate.

Each project is named after the process it models with the key having a fixed size of three characters, which correspond either the three first letters from the name of the process if it is composed by more than two words, not considering grammatical conjunctions, or by the first two letters of the first word plus the first from the second. The projects were created using the **"Simple Issue Tracking"** template which creates not just the project but also new Issue Type Scheme and Workflow Scheme. Through the creation of an Issue Type Schema and Workflow Schema we can restrict each project to use only the relevant issue types for the process and ensure they are handled in the correct workflow. Aside from these schemas that were automatically created for us, for each project we created a **Screen Scheme** which defines which fields appear in each screen and associates a screen to a transition and a **Type Screen Scheme** which associates an issue type with a screen schema.

Our **Field Configuration Scheme** was shared among all projects to allow for easily moving issues from one project to the next and provide us the flexibility to make fields required in one context but not on the other.

Notifications, which play a very important role in each process, were handled using a combination of the notification schemes and subscription to filters for things such as near SLA breaches. These subscriptions to filter work by performing the query specified in the filter and if any results are returned, i.e., if an issue breached its SLA, an mail will be sent to the groups or users that subscribed to that particular filter.

8.1 Incident Management

Incident Management are the first processes to be implemented as it is the most popular among the data we've collected. It is also one of the most straightforward to implement in SaaS CRM due to its nature.

Issue type:

By studying the definition of an incident in ITIL we were able to determine that an issue has, by default, fields that capture most of the information that is relevant for an incident such as priority, description and detailed activity log. In order to be fully compliant with ITIL's definition we created a new issue type, Incident, to which we added several custom fields. In Table below we detail the relevant fields, their type and what information they will store.

If during the lifecycle of the incident the user needs to assign tasks to a third party, he can do that by creating a subtask which will be linked to the originating incident.

Field	Type	Stored Information
Summary	Text field	Stores a single line description of the incident
Description	Multi-line text field	Stores a detailed description of the symptoms and the conditions where the incident was noticed
Priority	Priority	Details the priority of this incident
Event Category (custom field)	Cascading select list	Stores the category and subcategory of the incident, which can be either Hardware, Software or Network related.
Due Date	Date Time Picker	This field should be used when there is the need for defining a due date for the resolution of the incident.
Attachment	-	Allows the customer to attach files demonstrating the encountered incident.
Cause (custom field)	Multi-line text field	In this field the causes of the incident should be described.

Field	Type	Stored Information
Workarounds (custom field)	Multi-line text field	In this field the user should describe in detail the solution or workarounds for the incident
Labels	Label	This field should be populated with labels regarding the incident for posterior lookup
Line of support (custom field)	Single Select list	This field, which is updated automatically, stores the current line of support that is working on it
Reporter	User	Specifies who reported the incident
Assignee	User	Specifies to whom the incident is currently assigned
Participants	User list	List the users who can see the request in SaaS Service Desk
Request type	Request Type	Specifies the SaaS Service Desk request that generated the present incident or the Channel where it was reported

Workflow and Screens:

The workflow will define the steps in the lifecycle of the issue, so it is of the up most importance to have a flexible workflow that is capable to accurately represent each step where the incident will be and how it will progress while restricting the transitions that can be done in each step. To accomplish this, we used a mix of conditions, validators and post functions so that the user is only presented with valid options.

In the Figure 1.5 below we can see the workflow for incident management.

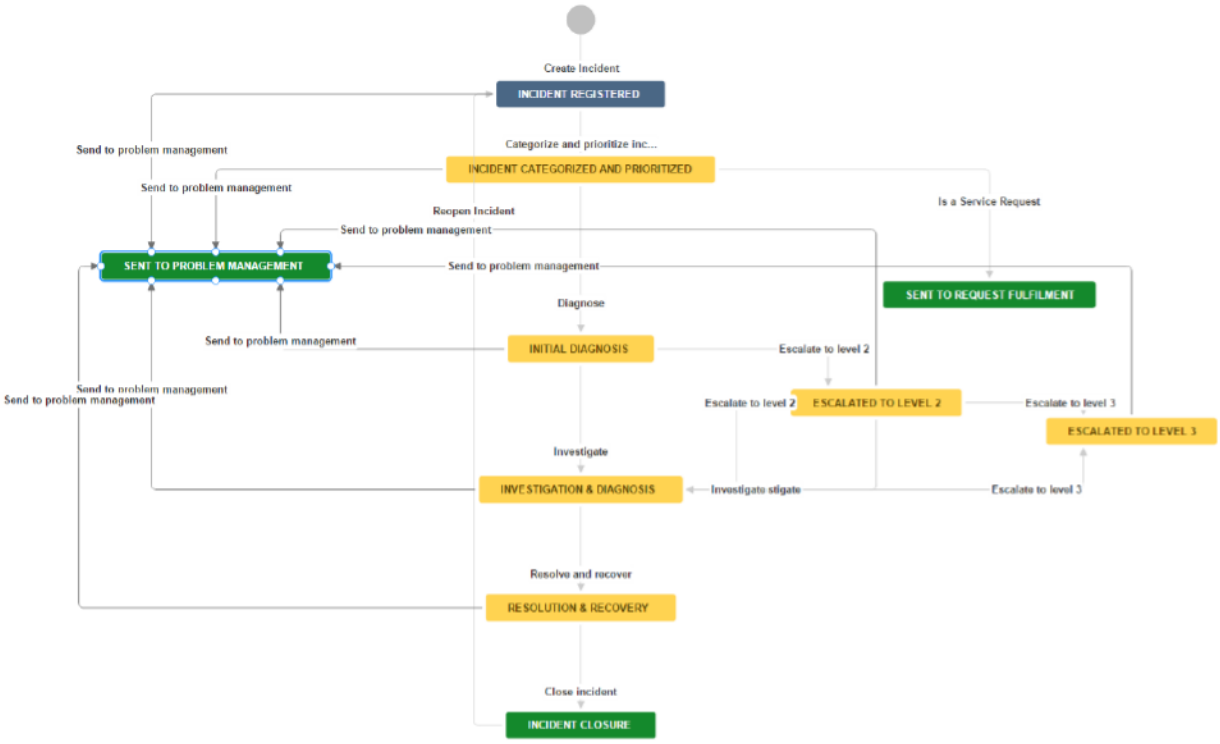

Figure: 1. 5 Incident workflow from SaaS CRM

An incident can be registered through CRM using the create issue menu or through SaaS Service Desk. Afterwards we categorize and prioritize the incident which is accomplished by showing a screen where the user must insert values for both fields, in case they haven't been previously filled. This transition uses a SaaS Suite Utilities validator to ensure that the category isn't set to "none" and a Script Runner validator that forces the priority to be a value different from "Not assessed". Once the incident is properly categorized, the user might discover that it isn't really an incident but in reality is either a change, problem or a service request. If it is the case, the user transitions to the correct state and through the use of Script Runner it is sent to the proper process, where a new issue will be created with a link to the originating incident and the original summary. Once the new issue is solved, the originating incident will be marked as done through the use of a post function provided by SaaS Misc Workflow Extensions which sets the resolution field from issues with a given link. If it is in fact an incident it can be moved into the investigation phase of the lifecycle where, after the user investigated the incident and its causes, can fill the cause and workarounds fields in case he managed to find the cause and solve the incident or escalate the incident to a more qualified team. The level to which the incident can be escalated is controlled by the Line of support field and SaaS Suite Utilities which only allow the incident to be escalated one

level at a time. Each transition to Investigation & Diagnosis state shows a screen where the user can fill the cause and workarounds.

Once the cause is found and the incident is solved the user can progress into the Resolve and Recover state. In this transition the user will be required to enter the cause and the workarounds in the screen by SaaS Suite Utilities and after the recovered is complete we can finally transition the incident to the final state where it will be marked as done.

8.2 Change Management

Change Management requires cooperation between both SaaS and Confluence in order to properly fulfill its goal. Here we will describe the work done in SaaS and in Knowledge Management we will describe in detail what exactly is a RFC and how it is achieved. The key for this process is CHM.

Issue type:

A change issue represents a single request for change, thus allowing the state or the RFC to be viewed by management. Despite this most of the relevant information is stored in a Confluence page detailing the RFC and linking to the issue. In Table 5.2 we present a list of fields present in a change issue.

Field	Type	Stored Information
Summary	Single-line text field	Stores a single line description of the change
Description	Multi-line text field	Stores a detailed description of the requested change
Change Priority	Priority	Details the priority of the change. This field uses different values than other issues in order to accommodate emergency changes.
Change Category (custom field)	Cascading select list	Stores the category of the change which can be either Hardware or Software and corresponding subcategories.
Due Date	Date Time Picker	This field should be used when there is the need for defining a due date for the change to be performed, in case it is approved.
Begin date	Date Time Picker	In this field we can store the scheduled date for the beginning of the deployment of the change.
End date	Date Time Picker	In this field we can store the scheduled date for the end of the deployment of the change.

Field	Type	Stored Information
Labels	Label	This field should be populated with labels regarding the change for posterior lookup.
Reporter	User	Specifies who requested the change.
Assignee	User	Specifies to whom the change is currently assigned.
Participants	User list	List the users who can see the request in SaaS Service Desk
Request type	Request Type	Specifies the SaaS Service Desk request that generated the present change or the Channel where it was reported

Workflow and Screens:

The change workflow, which can be seen in the Figure 1.6 below, offers two options right from the start, one for the small changes and other for changes that require more planning. We decided to use the same issue type for both, to the contrary of what was done in Request Fulfilment, due to small changes normally not being common enough to warrant the creation of a specific catalog.

For a trivial change the user has the possibility to bypass the normal workflow and head directly to the deployment of the change. This possibility is controlled via a SaaS Suite Utilities' condition in the transition which performs a query to see if the category of the change allows for a small change or not.

If the change isn't trivial, we perform the send request for review transition. In the resulting state the change should be reviewed by the user and either sent for approval or rejected. The decision is made through the transition selected by the user, in the former the add-on is used in order to obtain the approval of the CAB before it can progress into planning and in the latter the request's resolution is set to denied and moved into the terminal state Request denied.

In the transition for a small change and send request for review a post condition in the form of a script is ran which creates a new Confluence page using the summary and RFC id as title and copies the description and summary into the page.

When the RFC arrives at the Planning change state, the RFC page in Confluence will be update with all the available information. This can't be verified or enforced since we do not know of a way to check if the page has all the information. Before the RFC can transition into the next state SaaS Suite Utilities checks if the fields begin, end and due date have valid values, i.e., if the end and due date are posterior to the begin date.

After the change is deployed it needs to be reviewed. In order to ensure that the change isn't marked as complete without proper approval we used the Approval requiring the CAB to approve the change. If the one member of the CAB feels the change shouldn't be accepted, he can send the change back into the planning state. This transition is only visible to the members of the CAB group in SaaS CRM. Otherwise, if the change is approved it can be closed.

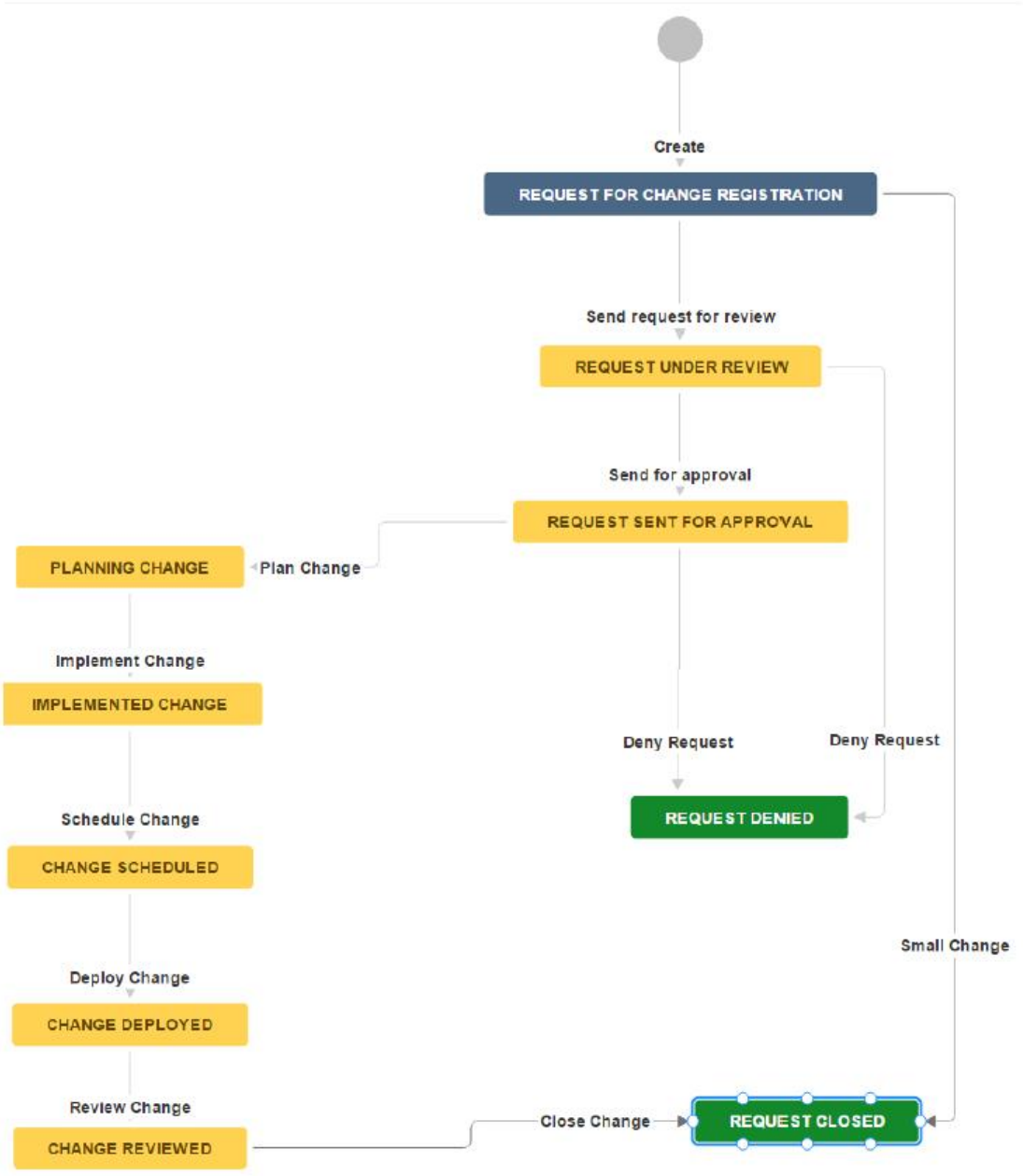

Figure: 1. 6 Change workflow from SaaS CRM

8.3 Request Fulfilment:

Request Fulfilment uses the acquired how-to from previous processes and combines them with SaaS capabilities to better serve the users and clients. The key for this project is REF.

Issue type:

During the implantation of this process, we divided the requests into two types, one for preapproved requests such as recovering the user's password and other for requests that need to be considered more carefully. We briefly considered using the same issue type for both and change path though the workflow using conditions and scripts but later we settled on using one issue type for preapproved requests and other for the remaining requests, each with their own workflow which are depicted in the Figures below respectively.

Workflow and Screens:

Requests can be registered thought the normal channels but unlike the previous processes depending on the type of request, the workflow will be different. We will begin by describing the workflow for preapproved requests and afterwards we'll talk about the normal request's workflow.

Preapproved requests are meant to be fulfilled quickly and with minimal intervention of the users. For this reason the preapproved request's workflow depicted in the Figure 1.7 below is so simple, containing only one state besides the mandatory created and closed. A request should be transition to the waiting for fulfilment once the user begins handling the request.

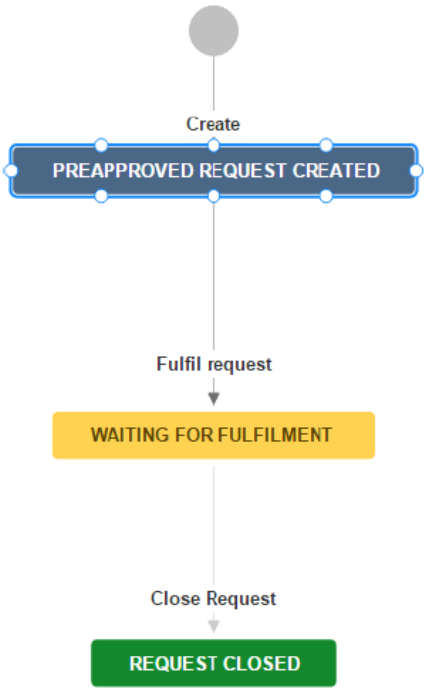

Figure: 1. 7 Preapproved workflow from SaaS CRM

For remaining requests, we use the workflow depicted in Figure 1.8 as follows. A request needs to be categorized before it can begin. Once again, we have a SaaS Suite Utilities validator ensuring the request has a category assigned and a priority different from 'Not assessed'. From this state forward the user can transition the request into Sent to Incident Management where a new incident is created with the summary and description of the originating request. Just like when a problem or request is created by and incident, the request will be marked as done when the new is closed.

Before a request can progress, it needs to be approved by the appropriate personnel which is achieved by the use of Approval add-on that requires the selected group or users to approve a transition before it can be executed. Once it is approved the request can be either escalated which increases line of support field or start handling the request itself which is done when the request is in the Waiting for fulfilment state. Once the request is complete it can finally be transition to Request closed where the resolution field is set to done.

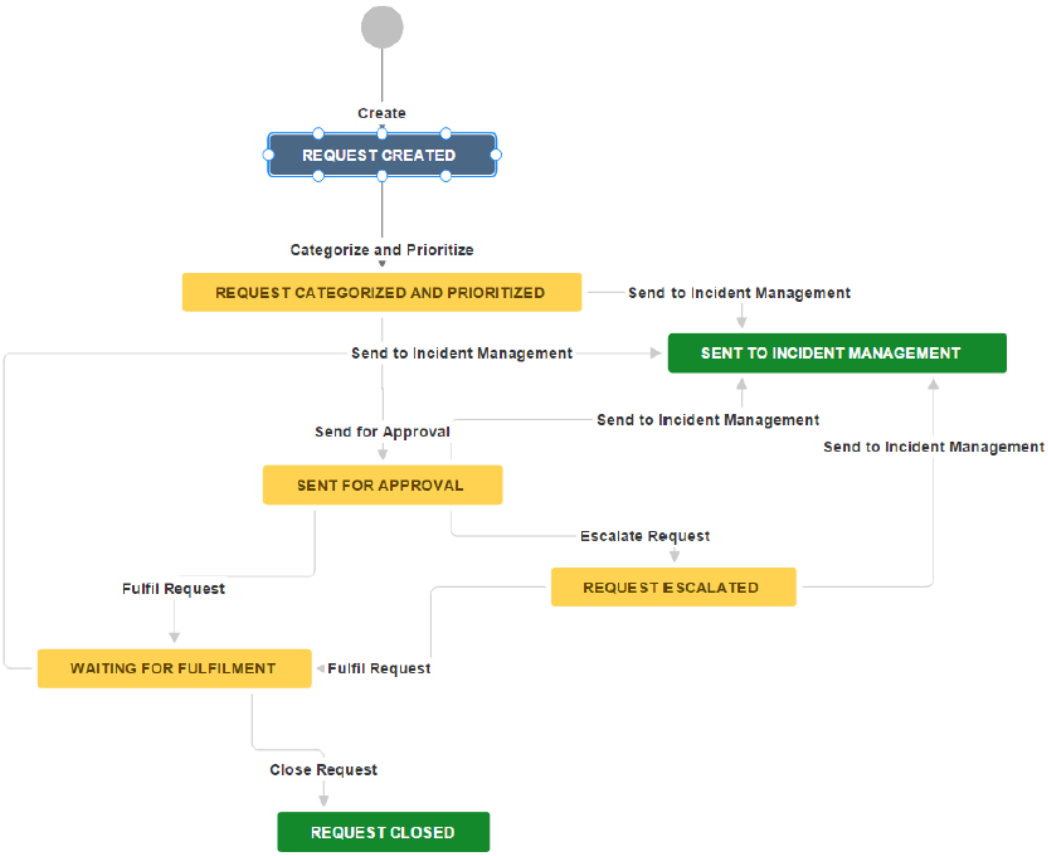

Figure: 1. 8 Request's workflow from SaaS CRM

8.4 Problem Management:

Project Management is implemented in a similar fashion to Incident Management as with the Known Error Database following the steps of Change Management. This process uses the Key PRM.

Issue type:

Due to similarities between an incident and a problem both issue types are similar with the exception of the field line of support being replaced by Review which is a multi-line text field where, at the end of the lifecycle of the problem, the user can review how the problem was handled and what can be done to improve. The key for this process is PRM.

Workflow and Screens:

Similarly to incidents, problem can be registered through SaaS CRM using the create issue menu or through SaaS Service Desk and later categorized and prioritized through the same screen.

Once the problem is properly categorized and prioritized, we move to the investigation and diagnosis phase. In this step the user should look in the **Known Error Database**, which is stored in a dedicated space in **Confluence**, for more information such as workarounds for the related incidents. If any matches are found, the entry is updated with the new information and the workaround is applied in order to restore at last some functionality while the problem remains open. If no match is found, a new entry in the KEDB is created based on the summary, and the investigation proceeds.

Once the cause and a solution is found, the KEDB should be updated and move into the **Resolution & Recover** state where the service should be restored to normal or sent to Change Management process to perform any functionality change.

After every major problem a review must be performed in order to assess what improvements can be introduced into the process to enable a quicker response for future problems. If the current problem is marked with major priority the only transition available is to major problem review state where the user must perform the review and introduce it in the review field. Once the review is stored in the issue, the related KEDB should be updated and the problem can be either closed or sent to Change Management.

The update must be done manually as we know of no way to either force the user to perform the update or do it automatically as shown in below Figure 1.9.

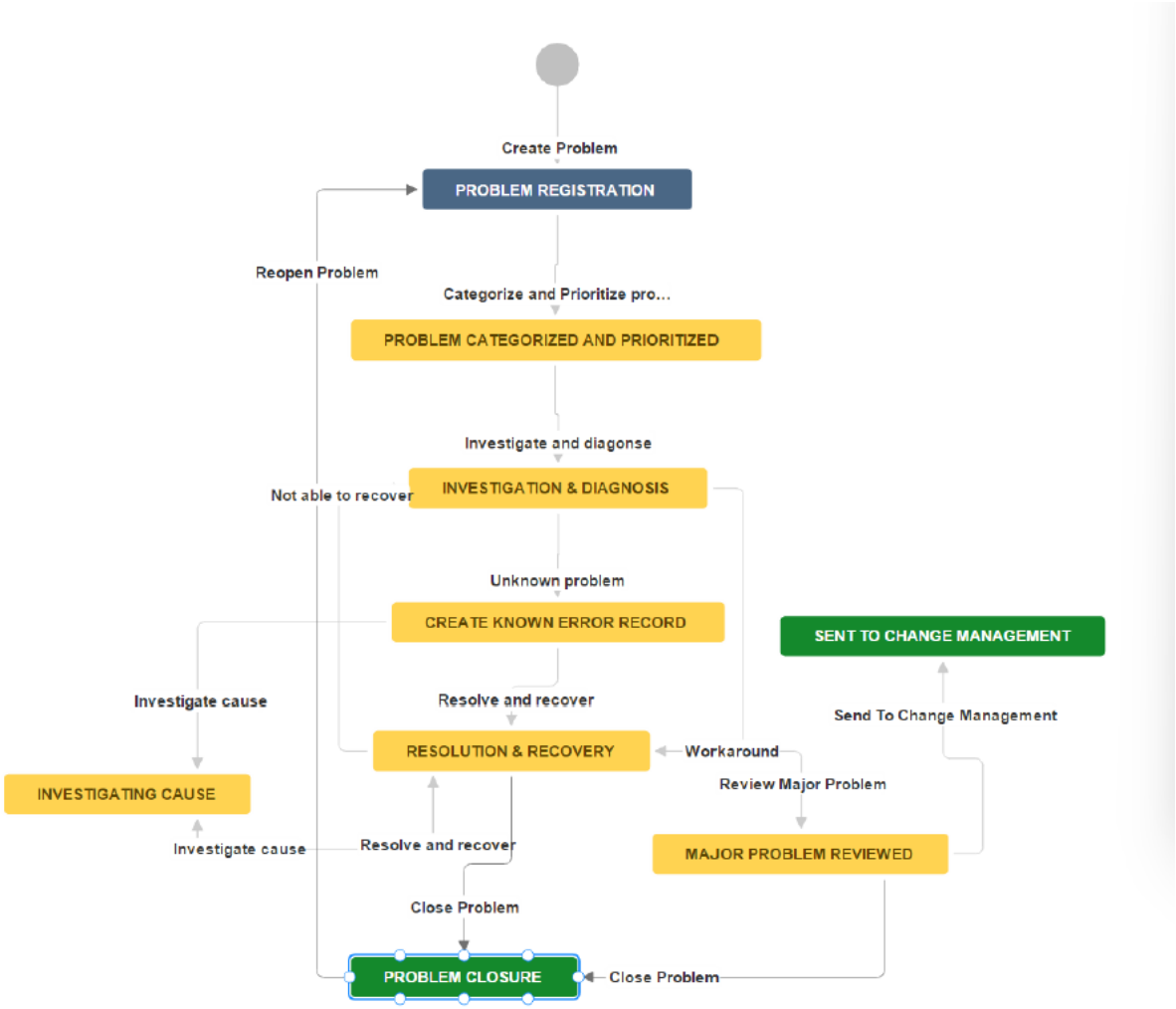

Figure: 1. 9 Problem's workflow from SaaS CRM

8.5 Event Management:

The main reason to implement this project was to assess how we could integrate, in the future, information from tools in our processes. Event Management was chosen particularly for the close relation it has with Incident and Problem Management. The key for this process is EVM.

Issue type:

Events are discernable occurrences that have value for management or service delivery, as such the number of events that are generated at any given time can be significant, but not all are equally relevant for service delivery. Some are just informational and don't require any action, others are warnings that the service is approaching a threshold and appropriate actions must be taken to prevent service disruptions and lastly there are the exception events where the service has been disrupted or degraded and immediate action is required. In order to prioritize the events we created an event significance that can be set to one of the previously referenced values and from there on the event will be handled accordingly.

Due to the close relationship between event management and incident, problem and change management we decided to copy the fields from the problem issue type and add the event significance. Fields such as cause or review can be used to store additional information about the origin of the event or additional notes for further improvement of the selected action.

In our artifact we created two channels for registering an event, one uses SaaS Service Desk where an event can be registered manually and the second is through automated emails from our monitoring tool.

Workflow and Screens:

Similarly to the other processes, the event management workflow begins by asking the user to provide a category and an event significance for the current event.

For exceptions we present the user with the possibility to create an incident, problem or change using Script Runner to create the new issue and link it to the originating event as shown in Figure 2.0.

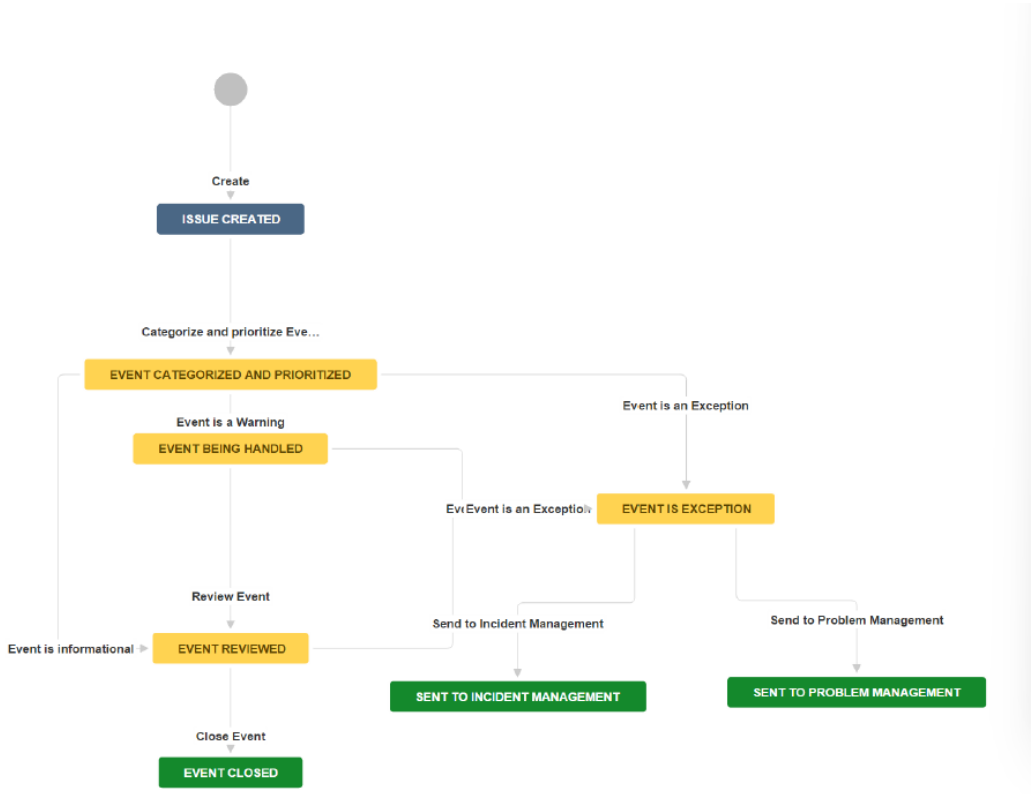

Figure 2.0 Event's workflow from SaaS CRM

8.6 Service Desk:

Service Desk consists in a functional unit that functions as a contact point between users and customers and as such it is responsible for resuming normal service to customers as soon as possible. Despite not having a project by itself, it was attributed the key SED.

In SaaS CRM, Service Desk's operations are managed by the aptly named SaaS Service Desk add-on. This add-on allows to create simple portals where the customer is given a set request available. Though this portal customers can also see the phase of the lifecycle of the issues they reported or are participants in and comment. In order to have a friendlier language it is possible to assign custom names to both the fields of the issue and the states of the workflow.

Once a request is closed, the reporting customer will receive an email asking for a quick rating of the service provided as well asking for any comments.

8.7 Service Level Management:

Service Level Management is responsible for negotiation the service's SLA and ensure it is fulfilled. It is also responsible for ensuring that incidents, problems and changes don't impact negatively the provision of the service. As such we felt it was very important to implement the parts used by the other processes as to ensure the each of the previously referred processes are handled in the appropriate time and don't put in jeopardy the provision of the service. This process was implemented mostly using SaaS Service Desk as it provides tools to monitor issues for SLA compliance and Confluence to store contracts. Despite not having a project by itself, it was attributed the key SLM.

Every time an incident, problem, request or change is created it is matched against the existing SLAs and assigned a time to resolution which will count down the time until the SLA is breached. SaaS Service Desk uses an internal query to select the issues to which each SLA applies. The combination of a filter to select the issues and a custom calendar setting allows to create flexible SLA that comply with the existing contracts.

In order to notify the relevant personnel about impending SLA breaches or actual breaches we created a couple of filters, one for breached SLAs and one for issues that will breach the SLA in less than 2 hours, for each process that looks for issues in these conditions and sends a mail to the group of users of the processes. The reason the notification goes to the group of users and not just the assignee of the issue in question is that for the mail be sent to just the assignee we would have to create a pair of filters for each user which would quickly become unmanageable.

For incidents we defined a two SLAs, one for incidents marked with blocker priority which should be handled within two hours while the remaining incidents should be handled within six hours. Both SLA are only applicable during business days from 9h to 18h. Similarly for requests we defined three SLA criteria, one for blocking requests which should be handled within 8 hours from the registration of the request. Preapproved requests and the remaining requests should be handled within 2 and 48 hours respectively business days, from 9:00 to 18:00.

Regarding problems, the SLAs are divided into three categories. The first relates to the problem acknowledgement where the time counts from the moment the problem was registered until it was categorized and prioritizes. Once it reaches that state, the target SLA is dependent on the priority of said problem. The last SLA group is related to the need, or not, for reviewing the problem. If it is necessary then the user has 2 days until closing the problem, otherwise he has 4 hours.

When and incident, problem, request or event is sent to another process, the originating issue is transitioned into a terminal state that despite not setting a resolution, stops the SLA clock. This exception is attributed to the fact that the recently created issue is subjected to a new SLA but the originating issue isn't resolved.

8.8 Knowledge Management:

Knowledge Management is entrusted with recording all relevant information. In our case it is concerned with Known errors and RFC. SaaS offers the possibility to create a Confluence page from a template and link it to the current issue but nor is it done automatically nor retrieves information from the issue to populate the page as such we decided to do this automatically resorting to Script Runner. Despite not having a project by itself, it was attributed the key KNM.

Known Error Database:

including the exact details about the symptoms and how they presented themselves and the exact steps taken to either resolve the problem or workaround it as to resume the service to its normal state. For this reason, it is very important that each entry is always up to date.

If a problem doesn't cause a serious service disruption and an effective workaround has been found and the cost of the permanent fix is to high compared with the returned value, the problem managers might decide to not resolve the problem and simply apply the workaround every time it is encountered. In this case the KEDB entry should be marked as such.

Users can create a new entry in the KEDB manually by proceeding to the Problem Management space in Confluence and selecting the Known Error entry in the list of available templates. When the entry is created automatically through the problem's lifecycle, the title is the summary, with the originating problem id in parentheses, and the problem section contains the description of the originating problem.

In order to reduce the number of registered incidents, this database (space) also has the solution to some common incidents. To ensure the Incident Management team only access the information about incidents, the articles which are intended to them have an "incident" label. This label allows us to filter the suggestions from SaaS Service Desk to just pages about incidents while still providing the Problem Management team with information about incidents and problems.

Change Database:

Each change should have its RFC stored in a Change Database that stores information regarding all changes, regardless of being completed, a draft or rejected. In each entry all the available information about the business case, risk and budget should be entered. If the RFC is rejected the reasons for doing so should also be detailed in the appropriated section.

Unlike the KEDB, it doesn't make sense to offer the possibility to create a new entry manually as an RFC that isn't connected to a change, in the correct process, won't ever be considered.

8.9 Add-ons:

In this subsection we will sum up the add-ons used in the implementation of each process and function, identified by their keys, and we will end with an estimation of the cost to acquire licenses for the add-ons needed for each process, for an organization with 500 to 3000 users with different types of editions in Figure 2.1. The cost is presented against add-ons per user and per month. In SaaS every user that should be able to create an issue is required to have a license, the same doesn't apply to SaaS Service Desk where only the agents are required to have a license (for both SaaS and SaaS Service Desk), customers are free. Agents in SaaS Service Desk are the personnel who interfaces with the customers and handles their requests. Every user will have access to Confluence as it is in it that the organization's Knowledge is stored. All prices refer to Server instances.

	Essentials € 25 /user/month*	Professional € 75 /user/month**	Enterprise € 150 /user/month*	Unlimited € 300 /user/month*
Account, Contact, Lead, and Opportunity Management	✓	✓	✓	✓
Email Integration with Gmail or Outlook	✓	✓	✓	✓
Salesforce Mobile App	✓	✓	✓	✓
Lead Registration and Rules-Based Lead Scoring	✗	✓	✓	✓
Collaborative Forecasting	✗	✓	✓	✓
Workflow and Approval Automation	✗	✗	✓	✓
24/7 Support and Configuration Services	✗	✗	✗	✓

Figure 2.1 SaaS Salesforce CRM Pricing per user by Edition

9. EVALUATION:

Despite the remarkable lack of available guidance in this area, evaluation is considered one of the most important steps of DSRM. It is in this phase that the resulting artifact from DSRM is validated against the previously defined objectives. Implementors, typically, have been left wondering how to properly use the available criteria and even which criteria should be used against their own artifacts.

As per our research the evaluation in DSRM should answer three questions: when, what and how. When refers to the time where the evaluation is performed, it can be either before or after the artifact is developed, ex ante or ex post respectively, or even at both times. What refers to subject of the evaluation, which can be either the Design Process or the Design Product. The first consists in a set of activities, tools, methods and practices used by the researchers while the latter consists on the tangible result of the applied process. Finally, how refers to evaluation: it can be either artificial where it is conducted under artificial conditions, or natural where the evaluation is conducted with real users in a real environment solving real problems. The when can easily be answered since our artifact is the instantiation, as such the evaluation was done ex post. For the what and how we based our evaluation on their dimensions, which are goal, environment, structure, activity and evolution, are closely related to the components of a design theory. The result can be seen in the Figure below. The goal, which specify the purpose of the artifact, can be evaluated by the efficacy of the result, the validity, which measures the correctness of the artifact, and the generality of the applicability of the artifact.

The environment is the set of people, organization and technology where the artifact will be evaluated. In this dimension the consistency of the artifact with the people is evaluated along with the utility, which measures how the artifact handles itself in the real world, the understandability, ease of use and ethicality. Consistency with organizations is evaluated along its ability to be integrated in the organization's existing environment, fit with organization and utility. Finally, consistency with technology evaluates if it uses the latest technologies. Both the consistency with the organization and technology evaluates the side effects of the artifact.

The structure of the artifact consists in the architecture that describes the artifact as such we should evaluate its completeness, simplicity, clarity, style, homomorphism level of detail and consistency. The activity dimension pertains to the functionality of the artifact that includes the completeness, consistency, accuracy, performance, and efficiency. Evolution pertains to the ability to withstand to changes in the environment, known as robustness, and its ability to improve its functionality based on its actions and responses from/to the environment as shown in Figure 2.2.

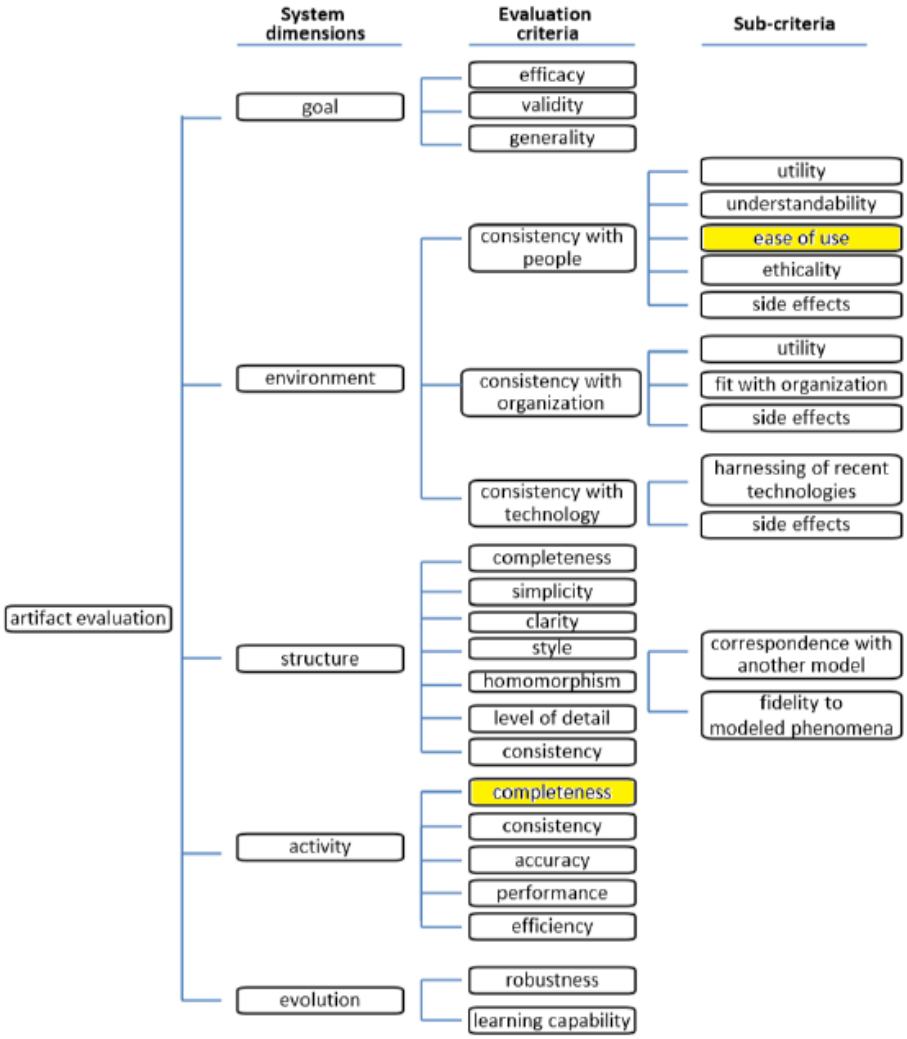

Figure 2.2 Evaluation Criteria Map

Based on the previously enunciated evaluation criteria and dimensions, we focused our evaluation on the completeness of the activity dimension.

Activity - completeness:

To evaluate the activity dimension, we chose to apply the completeness criteria which will be measured by checking the list of functional requirements for each implemented process as depicted in Figure 2.3. We chose not to evaluate Service Level Management as it wasn't considered for implementation, the only requirements that were implemented are in the context of the other processes.

Incident Management

Incident Management being one of the most straightforward to implement in a workflow management system like SaaS CRM is mostly implemented where it is supported. The only criteria not implemented are the ones related to Capacity Management, Availability Management and CI information resulting in 26/30 implemented requirements.

Change Management

In this process we managed to implement the base functionality, but we lack any type of risk analysis about the proposed/performance change. As such that information should be entered manually in each RFC stored in Confluence. Furthermore, each proposed RFC must be manually checked as to ensure that the necessary precautions are taken to prevent an OLA breach. Our current implementation doesn't support deployment of any kind, as such it should be handled externally and later acknowledged in our artifact. In this process we achieved 35/43 implemented requirements

Request Fulfilment

In this process we didn't implement 2 of the presented requirements. The first one being the ability to handle some requests automatically and the latter is the lack of billing support built in SaaS that prevents us from calculating the cost of a request. This process is rated at 18/20.

Problem Management

Problem Management is afflicted by the same lack of integration with Capacity Management, Availability Management and CI information that Incident Management is. As such we implemented 28/32 requirements.

Event Management

This is the process with less implemented requirements with only 9/15. This value is the result of SaaS not being able to monitor other systems, as such it isn't possible to define monitoring targets or thresholds.

Service Desk

The Service Desk function was completely implemented using SaaS Service Desk. The only requirements not covered by this add-on, performing user satisfaction survey and offering a FAQ, were readily done using Survey for Service Desk and Confluence respectively. It is implemented 9/9 requirements.

Knowledge Management

Knowledge Management was implemented using Confluence. The only requirement that isn't supported is the ability to present users with only approved pages. For this process we implemented 12/13 requirements.

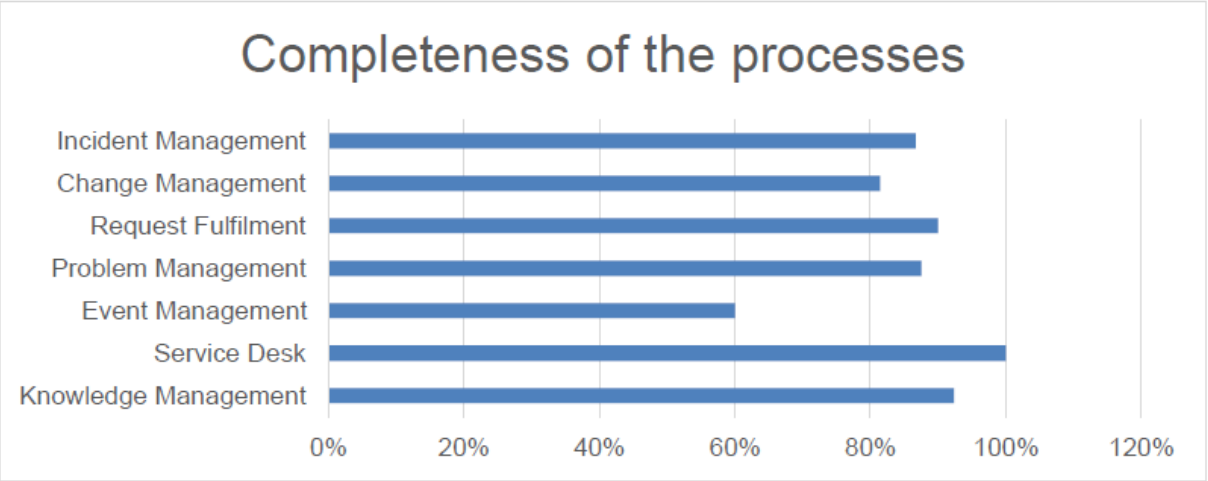

Figure 2. 3 Completeness of the processes

10. IMPLEMENTATION EFFORT:

It is not simply a marketing message, but also a fact that the technical effort for implementing SaaS technology is significantly lower than for traditional 'On Premise' solutions, assuming the necessary conceptual pre-work has been completed to the required level of detail.

SaaS implementation effort can be up to 50 per cent lower than 'On Premise' implementation related effort: the cost for application configuration can be lowered by half. Of course, these numbers depend on the complexity of processes and structures which need to be supported by the system. The more standard functionality is leveraged, the lower the final implementation effort is. However, for determining the total effort for the introduction of a new CRM system, the application configuration must be considered as well as all implementation-related activities. The following table in Figure 2.4 provides a detailed overview of the project activities and associated effort (per cent).

Project Phase	Effort (%) of Total
Requirements Verification & Definition	15
Application Setup & Customization	30
Interface Design & Implementation	10
Review Customization	5
Test & User Approval	5
Data Migration & Load	10
Training Support	5
System Go-Live	5
Shut-Down of Old SFA System	5
System Documentation	5
Dashboards/Reports	5
Sub-total	100
Contingency	20
Project Management	25

Figure 2. 4 Effort in per percent by project phase

11. BENEFITS:

SaaS CRM provides the standard functionality of On-Premise CRM solutions as well as the security, reliability and performance companies need to ensure smooth customer operations, without the time and cost associated with in-house systems. In a time where market requirements are changing companies no longer require CRM systems with predefined industry-specific or sub-industry-specific capabilities. They require a powerful tool that allows for rapid adoption to the market with reduced effort and resource involvement.

With the flexibility to adjust for rapid changing markets and the scalability to accommodate growth with the 'pay-as-you-go' concept SaaS is a huge leap forward. No longer purchasing new infrastructure or investing in ongoing system maintenance for the On-Premises system leads to overall cost containment. Zero-impact upgrades enable the environment to always be on the current technology.

Multi-tenancy – one essential attribute of Cloud Computing – allows everybody within the organization to access exactly the data he or she needs. Information can be easily shared within a controlled environment. Transparency of data, projects and processes is enforced. The CRM platform serves as a central data hub that increases consistency across all customer groups – the time of information silos has been superseded. Additionally, many SaaS tools provide internal collaboration functionality including companywide social networking, which accelerates the distribution of relevant information to specific recipient groups.

Easy configuration in real-time and extreme scalability provides banking sector with unique business flexibility, in particular when experimenting with today's advanced services and initiatives such as e-Banking, alternative sampling programs.

12. CONCLUSION:

Although the concept of Cloud Computing can satisfy an IT requirement and supports the business in acting as flexibly as the market requires, there are a few bottlenecks which may slow down the acceptance of SaaS and require close attention throughout the entire life cycle.

- Since data is stored on the vendor's servers, which is not necessarily located within the premises, data storage and handling needs to be evaluated from a legal perspective.
- To leverage the option for accelerated feature delivery, an appropriate governance and support model has to be in place before the first action towards implementation is taken. Since most SaaS applications are convenient and easy to customize, it is tempting to start implementing without considering the relevant organizational aspects up-front.
- Ideally, the governance model should include the support organization for the post deployment phase as well as a detailed SLA (Service Level Agreement) definition.
- Any implementation of a business application should be driven by the business, otherwise the project will probably trigger a software replacement initiative. Establishing the ideal contribution of business and IT to achieve overall quality for the new platform has proven to be an important factor for success.
- SaaS technology can deliver its benefits the more globally and harmonized the organization is set up. SaaS forces the organization to pursue a harmonization and standardization approach, looking at global processes and structures rather than local ones. Finding the balance between local and global alignment is one of the most challenging decisions the project sponsors must take, which is why this topic is often disregarded. The potential risks or challenges mentioned above can be mitigated by leveraging leading practices that have been established and developed during recent years. We have supported many clients moving from 'On Premise' to SaaS technology and has developed a proven methodology framework to support making the initiative a success – during and post-deployment.