



Implementação de Processos RH em SAP

FÁBIO CRISTIANO DA FONSECA RAMALHO

Outubro de 2016

Implementation of HR Processes in SAP

Fábio Cristiano da Fonseca Ramalho

**A dissertation submitted in partial fulfillment of
the requirements for the degree of Master of Science,
Specialization Area of Computer Systems**

Supervisor: Dr. Jorge Manuel Neves Coelho

Evaluation Committee:

President:

Members:

Dedictory

I dedicate this dissertation to my family for all the support I received during this project.

Abstract

Nowadays, HR management is a vital element for any business area in the world. For large scale companies, having a central tool to execute and approve all personnel data changes (including hiring and transferring employees) allows them to easily keep data up to date and reduce costs; all of this by using a global standard process.

The purpose of this dissertation is to prove how *SAP HCM Processes & Forms (HCM P&F)* can be used to create an interactive web application to manage all HR processes within a company (supporting any technology in the front-end) and how it compares to legacy applications that hold their own business logic instead of leveraging SAP ERP rules and functionalities.

Consequently, this document will describe in detail an implementation of *HCM P&F* developed by Konkconsulting, to show how it stands out in comparison to other application models. This implementation is currently used by a multinational corporation and supports over 10 types of HR processes with 200 fields.

Keywords: SAP, HR, P&F, Workflow, ABAP.

Resumo

Hoje em dia, a gestão de Recursos Humanos é um elemento fundamental para qualquer negócio no mundo. Empresas de média-grande magnitude necessitam de ferramentas centrais para gerir inúmeros processos (por exemplo: contratações, transferências e promoções), bem como facilitar as alterações de dados aos seus colaboradores. O facto de ter estes processos bem definidos permite comunicação rápida entre colaboradores, redução de burocracia e consequentemente, redução de custos.

Este documento descreve um modelo de desenvolvimento, orientado a SAP, para a criação de uma aplicação MSS/ESS que permita atingir os objectivos enunciados acima.

Para isto, será apresentada a *framework SAP HCM Processes & Forms (HCM P&F)* e demonstrado como esta pode ser utilizada para criar uma aplicação interactiva, que tire partido de todas as regras de negócio do ERP SAP – uma vantagem evidente sobre outros modelos de desenvolvimento. A solução apresentada está também preparada para suportar qualquer tecnologia de apresentação.

A implementação deste modelo foi levada a cabo pela Konkconsulting para uma multinacional com um modelo de Recursos Humanos complexo. A aplicação é neste momento utilizada por milhares de utilizadores e suporta mais de 10 tipos de processos.

Palavras-chave: SAP, Recursos, Humanos, P&F, Workflow, ABAP.

Contents

1	Introduction	1
1.1	Background	1
1.2	Problem	1
1.3	Value Analysis	2
1.4	Preconized Approach	2
1.5	Document Structure	3
2	Extended Background & Basic Concepts	5
2.1	SAP ERP	5
2.2	SAP Human Resources	6
2.2.1	Infotypes	7
2.2.2	Personnel Administration	7
2.2.3	Organizational Management	11
2.3	State of the Art in Technology	13
2.3.1	SAP HCM Processes & Forms	13
2.4	State of the Art in Solutions	15
2.4.1	Oracle Self-Service Human Resources	15
2.4.2	SuccessFactors Employee Central	18
2.5	Assumptions & Restrictions	20
2.6	Value Analysis	21
3	Solution Analysis	25
3.1	Personnel Change App (Legacy Application)	25
3.2	Enhanced SAP HCM Processes & Forms	27
3.2.1	Other Projects in SAP HCM P&F	29
4	Design	31
4.1	Functional requirements	31
4.1.1	Actors	31
4.1.2	Ticket Types	31
4.1.3	States	32
4.1.4	Use Cases	33
4.1.5	SAP HCM P&F Adaptation	33
4.2	Non-functional requirements	34
4.2.1	Performance	34
4.2.2	Scalability	34
4.2.3	Maintainability	34
4.3	Overall Architecture	34
4.3.1	API	35

4.3.2	SAP HCM Processes & Forms	35
4.3.3	SAP Workflow	36
4.3.4	Database	37
5	Implementation	39
5.1	Overview	39
5.2	API.....	39
5.2.1	Technical Overview	39
5.2.2	Primary functions	40
5.3	SAP HCM Processes & Forms	48
5.3.1	P&F Solution Overview	48
5.3.2	P&F Form Scenario	49
5.3.3	P&F Back-end Services	52
5.3.4	P&F Framework Enhancements	64
5.4	SAP Workflow	67
5.4.1	Technical Overview	67
5.4.2	Workflow Steps	71
5.5	P&F External Events	79
5.5.1	Technical Overview	79
5.5.2	Event Types	81
5.6	Code Review.....	83
6	Evaluation.....	85
6.1	User Satisfaction.....	85
6.2	Performance / Response Time	86
6.2.1	Phase 1	86
6.2.2	Phase 2	91
6.3	SAP System Usage	93
7	Summary	95
7.1	Results.....	95
7.2	Critical Reception	95
7.3	Future work	96
8	Appendices	102
8.1	Appendix 1 - Business Model Canvas	102
8.2	Appendix 2 - AHP	104
8.3	Appendix 3 - SAP HCM P&F Analysis	106
8.4	Appendix 4 - Use Cases	109

Figures

Figure 1 – Google Trends - ERP	5
Figure 2 – SAP Easy Access.....	6
Figure 3 – Enterprise Structure	8
Figure 4 – Personnel Structure – Employee Groups & Subgroups	9
Figure 5 – PA30 – Maintain HR Master Data	11
Figure 6 – Organizational Structure	11
Figure 7 – Organizational Structure – Jobs & Positions	12
Figure 8 – HCM P&F – Business Flow	13
Figure 9 – SAP Portal MSS – Start Process	14
Figure 10 – HCM P&F – Components.....	14
Figure 11 – Oracle SSHR – Manager Actions.....	16
Figure 12 – Oracle SSHR – Action Lifecycle	17
Figure 13 – SuccessFactors – Organizational Structure	18
Figure 14 – SuccessFactors Employee Central – Hire Employee.....	19
Figure 15 – Restrictions.....	20
Figure 16 – Value Analysis – Benefits vs. Sacrifices	21
Figure 17 – Value Analysis – AHP Diagram	22
Figure 18 – Value Analysis – AHP Decision Simulation	23
Figure 19 – Communication between Personnel Change App & SAP.....	26
Figure 20 – Personnel Change App vs. SAP Application.....	27
Figure 21 – SAP HCM P&F – Runtime Analysis.....	28
Figure 22 – State Diagram.....	32
Figure 23 – Use Cases Diagram	33
Figure 24 – Solution Architecture	35
Figure 25 – API Overview	35
Figure 26 – HCM Processes & Forms Overview	36
Figure 27 – SAP Workflow Overview	36
Figure 28 – Design Time Data Model.....	37
Figure 29 – Process Object Architecture.....	38
Figure 30 – Runtime Data Model	38
Figure 31 – API objects.....	40
Figure 32 – Get Form Skeleton – Flow of Execution	41
Figure 33 – Get Form Events – Flow of Execution	42
Figure 34 – Start Ticket – Flow of Execution.....	43
Figure 35 – Web Application – Start Ticket.....	43
Figure 36 – Ticket ready	44
Figure 37 – Process Event – Flow of Execution.....	45
Figure 38 – Process Standard Event – Flow of Execution	47
Figure 39 – HCM P&F Implementation Overview	48
Figure 40 – HCM P&F Process – YP_PCX.....	49

Figure 41 – HCM P&F Rule – Work Schedule	52
Figure 42 – HCM P&F – Back-end Services.....	53
Figure 43 – SAP_PA – Infotype 0002 configuration	54
Figure 44 – Generic Services Interfaces	55
Figure 45 – Generic Service: Additional Functions.....	56
Figure 46 – Generic Service: Date Specifications	59
Figure 47 – Infotype 0041 – Table structure	60
Figure 48 – Dates on the form.....	61
Figure 49 – Mapping between Infotypes and Multiple Records GS.....	63
Figure 50 – Generic Service: Multiple Records	63
Figure 51 – HCM P&F – Enhanced Form Check.....	65
Figure 52 – SAP Workflow Tasks	68
Figure 53 – SAP Workflow	69
Figure 54 – SAP Workflow Start	70
Figure 55 – HCM P&F Program Exit.....	71
Figure 56 – SAP Workflow Step – Condition “Is Manual Post”	73
Figure 57 – SAP Workflow – Assisted Posting	74
Figure 58 – SAP Workflow Step – Reset SAP Admin Decision.....	75
Figure 59 – SAP Workflow Step – Condition “Stop Assisted Posting”	77
Figure 60 – SAP Workflow Step – Condition “Close Assisted Posting”	78
Figure 61 – SAP Workflow Step – Condition “Complete Workflow”	78
Figure 62 – P&F External Events – Flow of Execution	80
Figure 63 – Full Search – Organizational Unit	82
Figure 64 – Code Inspector Categories	83
Figure 65 – Code Inspector Example	83
Figure 66 – Evaluation Timeline	85
Figure 67 – Transaction SAT – Example.....	87
Figure 68 – Transaction SAT – Example – Time per instruction	87
Figure 69 – Event – Cost Center Full Search.....	88
Figure 70 – ST05 – Activate SQL Trace	88
Figure 71 – ST05 – Cost Center Full Search Results	89
Figure 72 – Work Center Full Search – Box Plot.....	90
Figure 73 – Ticket creation – Box Plot.....	92
Figure 74 – HCM P&F Solution Analysis – Graphic.....	107
Figure 75 – HCM P&F Solution Analysis – Box Plot	108

Tables

Table 1 – Infotypes.....	7
Table 2 – SAP HCM P&F vs. Oracle SSHR	17
Table 3 – SAP HCM P&F vs. SFSF EC.....	19
Table 4 – Get Form Skeleton – Signature.....	40
Table 5 – Get Form Events – Signature	41
Table 6 – Process Event – Signature	45
Table 7 – Process Standard Event – Signature.....	46
Table 8 – HCM P&F Form – Relevant Infotypes	50
Table 9 – HCM P&F User Events	51
Table 10 – Generic Service Additional Functions – Default values.....	57
Table 11 – Generic Service Additional Functions – Possible values.....	58
Table 12 – Generic Service Additional Functions – Operations	59
Table 13 – ISR Structure – <i>BAPIRET2</i>	66
Table 14 – HCM P&F Structure – <i>HRBAS_MESSAGE</i>	66
Table 15 – SAP Workflow Start Event – Binding	70
Table 16 – SAP Workflow Step – Import Ticket Identifier	71
Table 17 – SAP Workflow Step – Edit & Submit Form Data.....	72
Table 18 – SAP Workflow Step – Import Ticket Effective Date.....	72
Table 19 – SAP Workflow Step – Execute Pre-Assisted Posting tasks	75
Table 20 – SAP Workflow Step – Assisted Posting.....	76
Table 21 – SAP Workflow Step – Invalidate Work Item.....	78
Table 22 – SAP Workflow Step – Revert SAP Post	79
Table 23 – <i>YPCX_CUST_EVENTS</i> – Structure	81
Table 24 – User Satisfaction – Observed vs. Expected	86
Table 25 – Work Center Basic Search – Execution Time Results	90
Table 26 – Ticket creation – Execution Time Results.....	92
Table 27 – SAP System Usage – Overview	93
Table 28 – <i>ISR_PROCESS_EVENT</i> Signature	106
Table 29 – HCM P&F Solution Analysis – Samples	107
Table 30 – Use Case 1 – Submit a New Ticket.....	109
Table 31 – Event Flow – Submit a New Ticket	109
Table 32 – Use Case 2 – Manage Compensation Data.....	109
Table 33 – Event Flow – Manage Compensation Data	109
Table 34 – Use Case 3 – Approve Ticket	110
Table 35 – Event Flow – Approve Ticket	110
Table 36 – Use Case 4 – Reject Ticket.....	110
Table 37 – Event Flow – Reject Ticket.....	110
Table 38 – Use Case 5 – Post to SAP	111
Table 39 – Event Flow – Post to SAP	111
Table 40 – Use Case 6 – Close Ticket	111

Table 41 – Event Flow – Close Ticket	111
--	-----

Acronyms and Symbols

Acronyms List

ABAP	Advanced Business Application Programming
AHP	Analytic Hierarchy Process
BAdIs	Business Add-ins
CRM	Customer Relationship Management
EBS	Oracle E-Business Suite
ECC	ERP Central Component
ERP	Enterprise Resource Planning
ESS	Employee Self-Service
FPM	Floor Plan Manager
GUI	Graphical User Interface
HCM	Human Capital Management
HR	Human Resources
ISR	Internal Service Request
MSS	Manager Self-Service
OM	Organizational Management
PA	Personnel Administration
PCR	Personnel Change Request
POC	Proof of Concept
P&F	Processes & Forms
RFC	Remote Function Call
SaaS	Software as a Service
SAP	Systems, Applications and Products in Data Processing
SAP SE	Official name of the company responsible for SAP ERP

SCM	Supply Chain Management
SFSF	Stock Quote for SuccessFactors, an SAP Company
SQL	Structures Query Language
SSHR	Oracle Self-Service Human Resources
UI	User Interface
UML	Unified Modeling Language
VP	Value Proposition

1 Introduction

The work described in this dissertation is related to the optimization of HR management in large scale organizations. Nowadays, the necessity for managing all these processes through computer applications is evident, in order to provide a clean and fast way of processing employee data. This chapter gives an overview of the area this dissertation focus on, by introducing the reader to the problem and the preconized approach.

1.1 Background

The world's labor market is constantly evolving, which forces companies to adjust and look for better solutions that help improve the quality of their services and reduce costs. In this process, HR management is an essential area to keep a company running their business successfully. In order to fulfill these requirements, companies normally turn to the IT market in search for a software that allows them to manage these processes, thus setting aside high-cost bureaucracy.

To answer these needs, ERPs emerged. This software provides a centralized and integrated vision of all the information in an organization (Garg & Venkitakrishnan, 2003, p. 3). Every module in an ERP is designed to help fulfil the majority of the requirements/problems organizations face. ERPs were built in a way that allow organizations to configure and customize the system at their own terms. On top of this, some companies choose to have external web applications that complement ERPs by expanding their features and dissociating end users from the complexity of the system.

This document presents a development model to cover these situations, focused on backend functionalities. It was implemented by Konkconsulting¹, "an innovative software development company, specialized in the design and implementation of processes in the area of Human Resources, with a focus on talent management" (konkconsulting, 2016), in an effort to answer a request from a customer regarding HR processes optimization, namely, new hires, promotions, organizational transfers, terminations and miscellaneous data changes. This customer uses SAP² ERP as the basis to hold all company data, therefore, the solution described in this dissertation is molded to this ERP.

1.2 Problem

The primary concern of this project is the creation of a centralized solution to facilitate personnel change requests in a large company, respecting its high standards. This company already owns a web application, built to accommodate their HR processes. Through this

¹ <http://www.konkconsulting.com>

² <http://go.sap.com/portugal/product/enterprise-management/erp.html>

application, users are able to request data changes, edit, approve or reject tickets, depending on their role in the organization. This application has been in use for almost 10 years and has suffered several changes along the years to fulfill new requirements, making it very difficult to maintain.

Taking this into consideration, the problem addressed by this dissertation relates to the process of finding the best solution to optimize this company's HR processes, leveraging the available tools and making it future-oriented. Should we improve the old application and develop new features on top of it? Is there an SAP tool that provides us the means to build an application, completely integrated with the ERP's business? Can we surpass the limitations of those tools or do we need to idealize an application from scratch? How can we guarantee that this is not just another version of the application, condemned to be replaced in a few years? How can we guarantee that, unlike the previous application, the one we design is scalable, customizable with low-effort developments, tolerant to change and easy for end users, without affecting performance?

1.3 Value Analysis

Managing people and their data is a complex process for any company with a reasonable workforce size. Employees, managers and administrators need an easy way of executing HR actions and escape all the bureaucracy. It is not uncommon to have an employee and a manager working in different cities or even different countries. With a solution as the one preconized in this document, the update of employee data or document approval is at the distance of a click. For that reason, this solution is ideal for companies struggling with the management of HR data, who want to standardize their processes and reduce costs. Furthermore, much of the value of this solution is in the fact that it is designed to be easily configurable and customizable, which makes the response to new features a fast process.

The value analysis of the solution will be addressed in detail in section 2.6 Value Analysis under chapter 2 Extended Background & Basic Concepts.

1.4 Preconized Approach

As enumerated in section 1.2 Problem, there are several approaches to consider when implementing an HR application based in SAP systems. Normally, the first option is to take advantage of SAP standard frameworks/tools to produce the desired effect. Evidently, sometimes, standard features do not cover all the requirements customers establish. In that case, there's two options: go for a hybrid scenario where we are able to complement standard features with custom ones or develop a completely new application.

The preconized approach of this project is to analyze the flexibility of a standard framework – *SAP HCM Processes & Forms* – and along with custom enhancements, achieve the desired HR management application.

1.5 Document Structure

This dissertation is structured to provide the reader with a level of knowledge and detail proportional to their advance in the document, which means high-level concepts will be presented first and full solution details later.

This first chapter presents the subject of this dissertation, including a brief description of the problem at hand, its value and the approach to solve it.

In the second chapter, 2 Extended Background & Basic Concepts, this document shows all the concepts and business details necessary to fully understand the solution. It also contains an analysis on the value of this product as well as the state of the art in equivalent solutions/technologies.

The third chapter, 3 Solution Analysis, contains details on the process of finding a solution, including the methods used to evaluate all possible approaches.

The fourth chapter, 4 Design, as the name suggests, contains the design of the preconized solution, with the use of high-level diagrams to explain the business behind each functionality.

The fifth chapter, 5 Implementation, takes the reader through the implementation process, including all objects that were built and how they combine to produce the solution.

The sixth chapter, 6 Evaluation, details the experiences and tests that were done to assure the success of this solution.

The seventh and last chapter contains the conclusions drawn from this work.

2 Extended Background & Basic Concepts

In this chapter, there will be a global contextualization of the project. It contains an overview of the SAP ERP and HR concepts required to understand the solution. Additionally, there's a section dedicated to the state of the art in equivalent solutions.

2.1 SAP ERP

SAP ERP is the primary product of German company SAP SE³. This ERP “is a proven, trusted foundation – built to support companies of all sizes across all industries. Leverage role-based access to critical data, applications, and analytical tools – and streamline your processes across procurement, manufacturing, service, sales, finance, and HR” (SAP SE, 2016a).

SAP has been the world leader in ERP market for the past few years. For instance, in 2013, SAP had a 24% market share, according to Gartner⁴.

Furthermore, SAP is at the top of ERP related queries, as shown below by Google Trends:

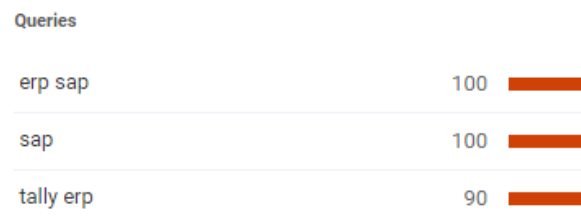


Figure 1 – Google Trends - ERP

SAP ERP consists of several modules, each one valid for a specific area of specialization in a company. The list below contains all the modules provided by SAP ECC (the current version of SAP ERP):

- CO – Controlling
- FI – Finance
- **HR – Human Resources**
- MM – Material Management
- PM – Plant Maintenance
- PP – Production Planning
- PS – Project System
- QM – Quality Management
- SD – Sales & Distribution

³ <http://go.sap.com/>

⁴ <http://www.forbes.com/sites/louiscolumnbus/2014/05/12/gartners-erp-market-share-update-shows-the-future-of-cloud-erp-is-now/>

Users access SAP ERP through SAP GUI⁵. SAP GUI is a client application that allows people to access different SAP systems. Once authenticated, users are re-directed to the ERP's primary screen – SAP Easy Access:

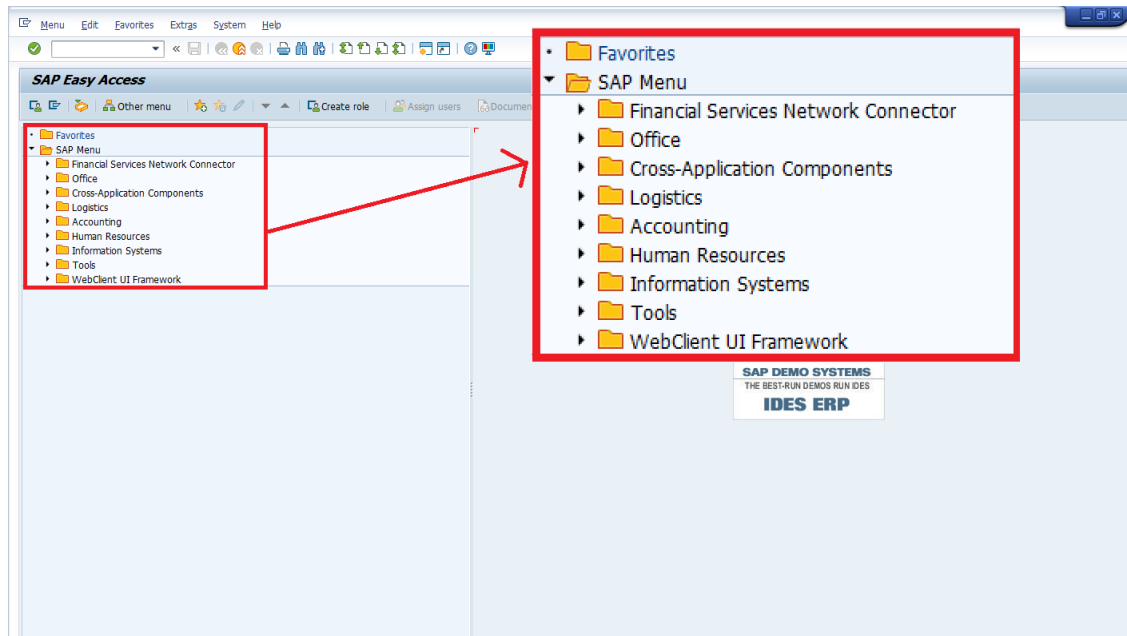


Figure 2 – SAP Easy Access

Each node in the picture is comprised by several levels of sub-nodes, until it reaches the final object: a **transaction**. A transaction consists in a set of SAP screens. In other words, it is an application inside the ERP. They are the destination point of user navigation. Even though all transactions can be accessed through SAP Easy Access, each one has an associated code – a transaction code. Hence, the toolbar contains a box that works like a shortcut to any transaction.

2.2 SAP Human Resources

As described earlier, the purpose of this work is to develop a new model to optimize HR processes. For that reason, this document focuses on SAP HR, specifically, submodules:

- *Personnel Administration (PA)*
PA is a central repository for managing employee information (HR Master Data) (Krämer, et al., 2006, p. 39).
- *Organizational Management (OM)*
OM is a central tool to manage a company's organizational structure, the basis for personnel planning and development processes (Krämer, et al., 2006, p. 141).

⁵ <http://scn.sap.com/community/gui>

2.2.1 Infotypes

Infotypes concept is perhaps the most relevant in the area of SAP Human Resources. Infotypes are “a combination of professional data which belongs together, e.g., addresses, bank details, additional payment, etc.” (Krämer, et al., 2006, p. 54). Infotypes are identified by a number from 0000 to 9999 (Krämer, et al., 2006, p. 54). For instance, infotype *Personal Data* (0002) contains the unique characteristics of an employee: name, date of birth, country, nationality, etc.

Infotype Number	Reserved for
0000-0999	HR Master Data
1000-1999	Organizational Management
2000-2999	Time Management
4000-4999	Recruitment
9000-9999	Customer range

Table 1 – Infotypes

Infotypes may also be divided in **infosubtypes**. An example is infotype *Family/Related Person* (0021), which has infosubtypes *Father* and *Child*. In the same way, a group of infotypes related to each other are called **infogroups**.

Additionally, there's the concept of **time constraint**, an essential attribute of the infotype (Krämer, et al., 2006, p. 55). Time constraint “allows HR administration to build a history of data that changes with time. Time constraint describes to what extent an infotype or subtype can exist on a multiple basis and can contain gaps.” (Krämer, et al., 2006, p. 55).

These are the most common time constraint types:

- 0 – Exactly one record must exist during the entire period of validity of the person (Krämer, et al., 2006, p. 56);
- 1 – For one point in time, there must be exactly one valid record and overlaps are not possible (Krämer, et al., 2006, p. 56);
- 2 – For each point in time, there can be a maximum of one valid record. Gaps are allowed but no overlaps (Krämer, et al., 2006, p. 56);
- 3 – No restrictions (i.e. there may be several records or none for a point in time).

2.2.2 Personnel Administration

SAP introduces Personnel Administration as a central tool to hold personnel data. “In times of increasing decentralization and the globalization of markets, a central and constantly accessible administration for personnel data is becoming an increasingly decisive advantage. Personnel Administration relieves you from the daily administrative routine activities which are costly and time consuming, while simultaneously supporting you in demanding planning activities” (SAP SE, 2016b).

The administrative tasks provided by this module are the following:

- Save employee data in infotypes (HR master data);
- Execute *personnel actions* to administer HR master data, such as hiring an employee;
- Mass update of employee data (i.e., *Master Data Fast Entry*);
- Archive and administer documents;
- Automatize actions subsequent to data changes, through *Workflow-Management*;
- Allow employees to edit their own data – ESS (*Employee Self-Service*).

The existing functions for HR master data maintenance (infotypes 0000-9999) allow users to insert, update and display employee data. In PA, this is structured through the organizational assignment, which is divided in three structures (SAP SE, 2016c):

- Enterprise Structure
- Personnel Structure
- Pay Scale Structure

Each employee in the organization is unique, i.e., he/she has a person ID, belongs to a specific work location and has a specific employment status. This kind of data is saved in infotype *Organizational Assignment* (0001).

2.2.2.1 Enterprise Structure

Enterprise Structure is a “structure of the company according to personnel administrative, time management and payroll perspectives from the point of view of your own company” (SAP SE, 2016d).

This structure consists of 4 levels:

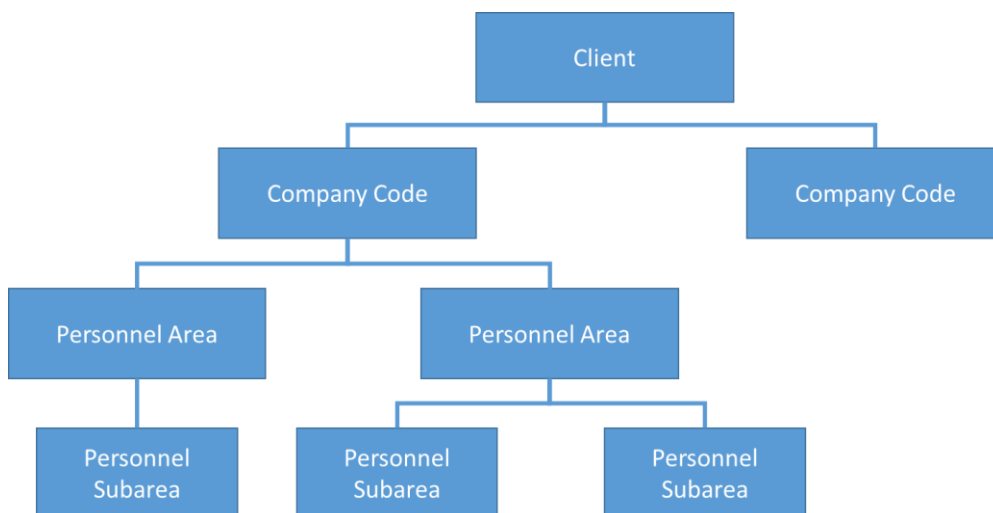


Figure 3 – Enterprise Structure

Client corresponds to an isolated unit within the SAP system. A client’s area of jurisdiction can correspond to a unit as small as a company or as large as an entire enterprise (SAP SE, 2016d).

Company Code “is the smallest organizational unit of external accounting in which a complete and isolated financial accounting can be created” (SAP SE, 2016d). Normally, company codes are associated with one or several countries that forces the elements below to belong to the same countries.

A **Personnel Area** represents a delimited enterprise area. It is used as selection criteria for authorization checking (SAP SE, 2016d).

Personnel subareas are a sublevel of personnel areas that help define more specific regulations for employees. They can be legal or company specific regulations.

2.2.2.2 Personnel Structure

This structure “describes an employee’s position in a company from the individual employee’s view” (SAP SE, 2016e).

It can be considered from two perspectives (administrative and organizational). This section will focus on the administrative perspective, since the organizational structure will be later described under 2.2.3 Organizational Management.

The administrative personnel structure is divided into:

- **Employee Group**
 - “Employee groups represent a primary subdivision of personnel. An employee group defines the extent to which its employees place their labor at the disposal of the enterprise. The Personnel Administration component makes a significant differentiation between active, pensioner and early retiree employee groups” (SAP SE, 2016e).
- **Employee Subgroups**
 - Employee subgroups subdivide employee groups. Within the employee group for active employees, for example, a distinction is made between hourly-paid and monthly-paid employees (SAP SE, 2016e).

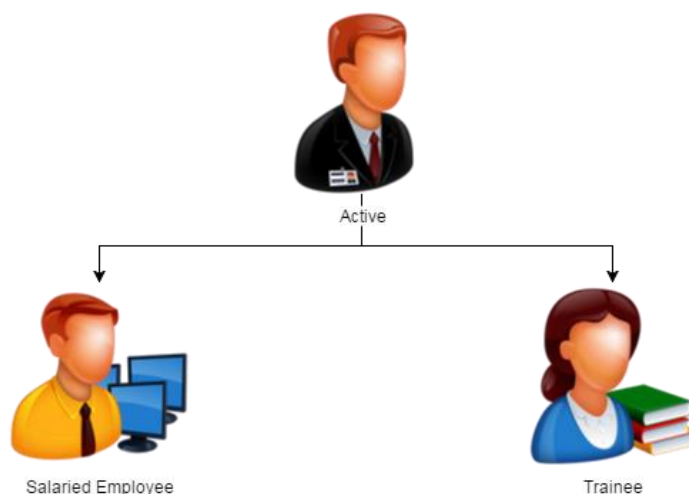


Figure 4 – Personnel Structure – Employee Groups & Subgroups

- **Payroll Area**
 - The payroll area is an organizational unit in the Human Resources department (SAP SE, 2016e) which allows companies to group employees that should have their salaries processed similarly.
- **Organizational Key**
 - “The organizational key consists of a 14-character field” (SAP SE, 2016e) that may be used to define the position of the employee in more detail.

2.2.2.3 Pay Scale Structure

This structure, as the name suggests, has “the primary aim of determining each remuneration according to collective agreement” (SAP SE, 2016f) and consists of:

- **Pay Scale Type**
 - “The pay scale type defines the area of economic activity for which a collective agreement is valid” (SAP SE, 2016g).
- **Pay Scale Area**
 - “The pay scale area defines the geographical area in which a collective agreement is valid” (SAP SE, 2016g).
- **Employee Subgroup groupings of Collective Agreement**
 - “With the employee subgroup grouping for collective agreement provision you restrict the eligibility of pay scale groups, so that only certain pay scale groups are valid for specific employee subgroups” (SAP SE, 2016g).
- **Pay Scale Group & Level**
 - “Pay scale groups and pay scale levels are the criteria used to classify data for job evaluations and indirect valuations. Pay scale levels are subdivisions of pay scale groups” (SAP SE, 2016g).

2.2.2.4 Master data maintenance

Now that these structures were presented, it’s time to address how they are handled.

As already mentioned, there’s the concept of **personnel actions** which map directly to infotype *Actions* (0000). Personnel Actions combine several infotypes that are related to each other, making it easy for users to enter all the data that the system requires. For instance, the action of hiring an employee, which prompts the user to enter a large amount of data, like personal data and the assignment of the employee to the organization.

From a functional point of view, there are 3 transactions in SAP to access this data:

- **PA20** – Display employee master data
- **PA30** – Maintain employee master data
- **PA40** – Execute personnel actions

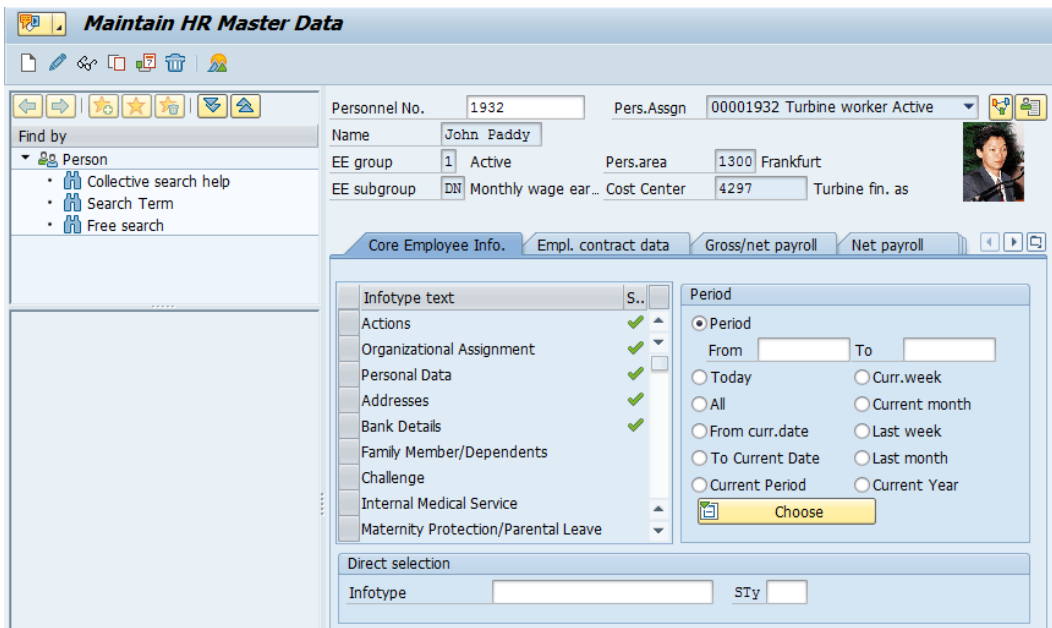


Figure 5 – PA30 – Maintain HR Master Data

2.2.3 Organizational Management

Every company needs an organizational structure. It defines the place of each employee in the company. Organizational Management (OM) is SAP's HR module built for that purpose. Considering that the purpose of this project is to build an HR management application, it is through this module that several searches will be made. Figure 6 shows the organizational structure within SAP.

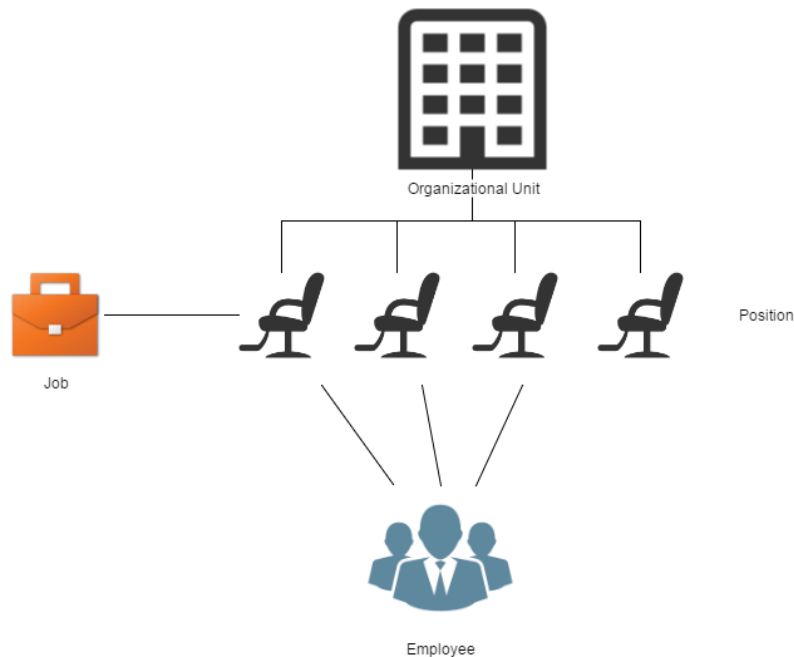


Figure 6 – Organizational Structure

“**Organizational Units (O)** are functional units in an enterprise” (SAP SE, 2016h). Normally, they are the departments of a company. They can be subdivided into other organizational units, according to geographical aspects or project teams.

Jobs (C) are a general classification for a predefined set of tasks. According to SAP, “jobs serve as job descriptions that apply to several positions with similar tasks or characteristics” (SAP SE, 2016i).

“**Positions (S)** are concrete and are held by persons in an enterprise” (SAP SE, 2016i). Positions inherit the characteristics of a job. See below an example:

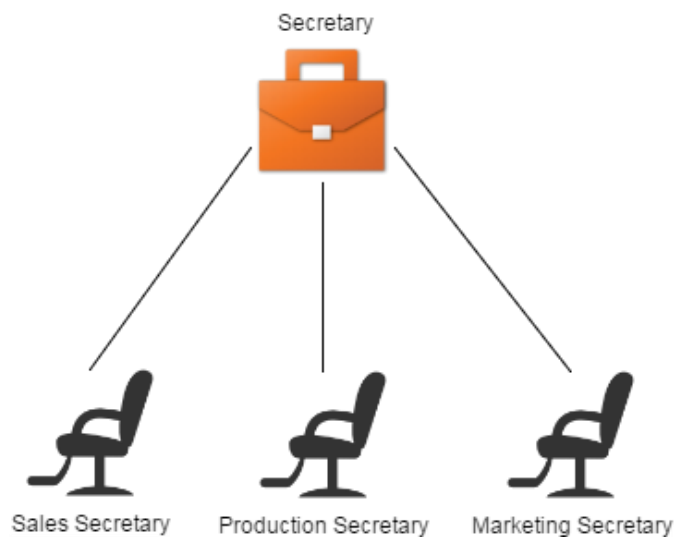


Figure 7 – Organizational Structure – Jobs & Positions

Each position is assigned to a **Work Center (A)**, which identifies the physic location where the employee works.

In SAP ERP, an employee is an object type **Person (P)**.

As seen in Table 1 – Infotypes, OM infotypes are placed between 1000 and 1999. From this list, it’s important to highlight two:

- 1000 – Object – Holds object characteristics. Positions names, etc. are saved here.
- 1001 – Relationships – Holds relationships between objects. E.g. position to job.

2.3 State of the Art in Technology

In this section, one can find a brief analysis on the state of the technology used to implement the desired solution.

2.3.1 SAP HCM Processes & Forms

Organizations are in constant pursuit of increased speed of data processing. Getting an employee on-boarded or their promotion completed in a timely manner is an obvious business imperative (Morgalis & Toombs, 2013, p. 50). Freed of the responsibilities of intensively manual paper processes, HR can devote time to its core purpose: ensuring that the organization's workforce is capable of supporting the strategic objectives of the business (Morgalis & Toombs, 2013, p. 50).

To accomplish these objectives, SAP provides **HCM Processes & Forms**, a framework to create, manage and execute data-intensive HR processes through interactive forms, which integrate all involved roles in the process (SAP SE, 2016j).

The following picture shows a typical flow of an HCM P&F process:

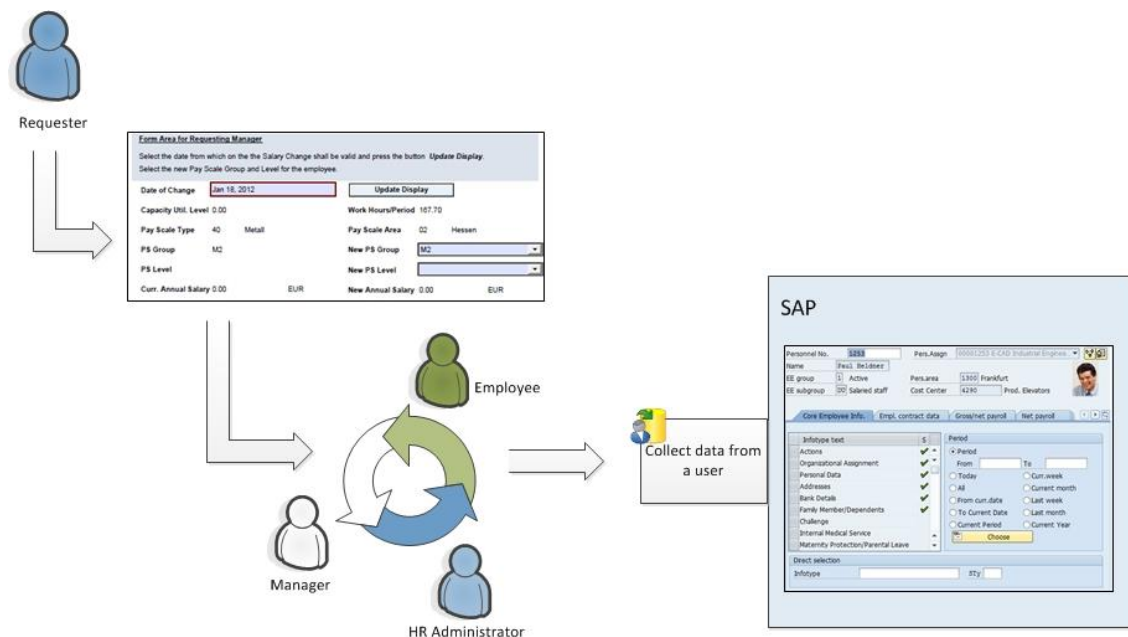


Figure 8 – HCM P&F – Business Flow

HCM Processes & Forms is embedded into SAP MSS/ESS applications, part of SAP Portal solution (SAP SE, 2016o). In fully standard HCM P&F scenarios, users access the portal to process HR actions (portal applications are dynamically designed according to the configuration setup in HCM P&F).

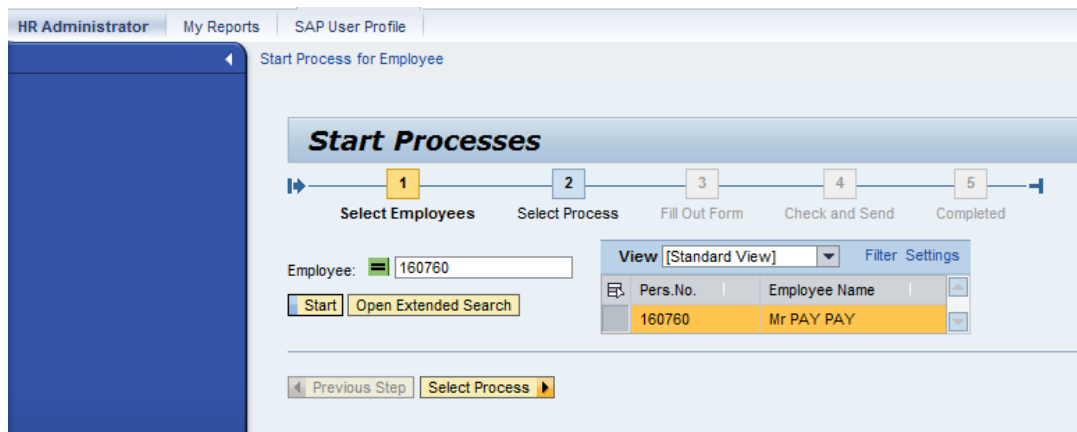


Figure 9 – SAP Portal MSS – Start Process

From a design point of view, HCM P&F is setup through both configuration and custom developments. There are three major components:

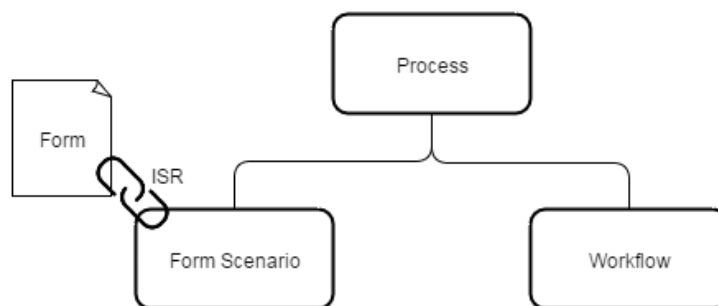


Figure 10 – HCM P&F – Components

Process is the parent component, which aggregates all of the others. “From a business point of view, the process is the basis for the implementation of a process. Therefore, the business flow should exist in the form of a process description at the start of the implementation” (SAP SE, 2016l). “From an implementation point of view, the process is the technical representation of the business point of view using Customizing and using elements from HR Administrative Services” (SAP SE, 2016l).

Form Scenario is the component used to set the form fields, configure default values, input helps and the assignment of those fields to infotypes (SAP SE, 2016m). This is complemented with configurable user events and rules to turn on/off functionalities. Connected to the form scenario (through ISR), there’s the form itself (UI design), which currently supports four types:

- Adobe Form
- FPM Form (based on Web Dynpro⁶)
- Roadmap Form (based on Web Dynpro)
- Mass Form (based on Web Dynpro)

⁶https://help.sap.com/saphelp_nw70/helpdata/en/77/3545415ea6f523e1000000a155106/frameset.htm

For many years, Adobe Forms was the only option. Web Dynpro was introduced to HCM P&F in 2012 (Rajora, 2012). Nowadays, Adobe Forms are clearly an outsider. Besides the limited layout, license costs are enormous. Adobe licenses are requested **per form per user** (Solomon, 2009) – a major setback for big corporations.

The **workflow** is used to specify which steps should follow in the process and who is responsible for processing/approving them (SAP SE, 2016n).

In addition to the referred components, from a technical point of view, HCM P&F is equipped with Back-end Services / Generic Services. These last services allow developers to add custom business logic (for example, new validations) (SAP SE, 2016p).

2.4 State of the Art in Solutions

In this section, you can find information on two more HR market solutions. The emphasis will be their core HR components and how they allow companies to setup business processes.

2.4.1 Oracle Self-Service Human Resources

Oracle Corporation⁷ is a global computer technology corporation. For more than three and a half decades, Oracle has been the leader in database software (Oracle Corporation, 2016a). Oracle is also a strong supplier in the ERP, CRM and SCM markets.

One of Oracle's offerings in ERP market is Oracle E-Business Suite (EBS), a fully integrated, comprehensive suite of business applications (Oracle Corporation, 2016b). EBS is divided into 9 modules: CRM, Service Management, Financial Management, **Human Capital Management**, Project Portfolio Management, Advanced Procurement, SCM, Value Chain Planning, and Value Chain Execution.

Within HCM module, Oracle provides several applications. For instance, Self-Service Human Resources (SSHR), iRecruitment and Performance Management. For the purpose of this dissertation, we'll focus on SSHR.

Oracle SSHR “offers secure self-service business transactions and easy-to-use functionality driven by Oracle's proven web and workflow technologies” (Oracle Corporation, 2015, p. 1:1) SSHR key features include configurability, dynamic routing of transactions and streamlined business processes (Oracle Corporation, 2015, p. 1:2), crucial functionalities to the setup of company-specific HR processes.

From a user point of view, SSHR provides both ESS and MSS functions. Through Self-Service Actions – business processes that change the conditions of employment in the enterprise

⁷ <http://www.oracle.com/index.html>

(Oracle Corporation, 2015, p. 11:43) –, employees are able to execute **manager actions** or **personal actions**:

- Manager Actions “enable users to first select a person and then the function, or action, to be performed” (Oracle Corporation, 2015, p. 1:8);
- Personal Actions “present users with a context-sensitive list of actions that they can perform on themselves” (Oracle Corporation, 2015, p. 1:9).

Figure 11 shows an example of a manager action (promotion), using SSHR (Oracle Corporation, 2014).

The screenshot displays the Oracle SSHR Manager Actions interface. On the left is a navigation menu with options like 'Worker Status Change', 'Change Hours', 'Termination', 'Change Pay', 'Change Job', 'Promotion', 'Change Cost Center, Location and Manager', 'Personal Information', 'Tenure Status', 'Transfer', 'Education and Qualifications', 'Other Professional Qualifications', 'Work Preferences', 'Change Cost Center, Terms and Manager', 'Extra Information', 'Change Worker Status and Terms', 'Employee Review', 'Individual Compensation Distributions', and 'Private Vehicle Entry'. The 'Promotion' option is selected. The main area is titled 'Promotion : Assignment' and contains the following information: Employee Name: Bennett, Terry; Organization Email Address: [blank]; Manager: Palmer, Blair; Position: MGR519.Recruiting Manager; Assignment Category: Fulltime-Regular; Context: [blank]. Below this, there are two radio buttons for effective date: 'Changes should take effect on the effective date as entered below.' (selected) and 'Changes should take effect as soon as final approval is made.' The 'Effective Date' is set to '14-May-2013'. A note states '* Indicates required field'. The 'Department' section shows: Department Name: Corporate Human Resources; Address Line 1: 90 Fifth Avenue; City: New York; State: NY; Zip Code: 10022-3422; County: New York; Country: United States. The 'Job' section shows Job Title: CON600.Consultant.

Figure 11 – Oracle SSHR – Manager Actions

The list of actions available to managers/employees can be limited through an eligibility process (Oracle Corporation, 2015, p. 11:34).

The approval workflow behind these actions is also configurable. Companies may set up rules that define a list of approvers/reviewers for each action. To empower that configuration, Oracle supplies three overall stages (Oracle Corporation, 2015, p. 11:44): Initiate, Approve, Apply.

Each one has a range of options and features to configure unique process flows (Oracle Corporation, 2015, p. 11:45):

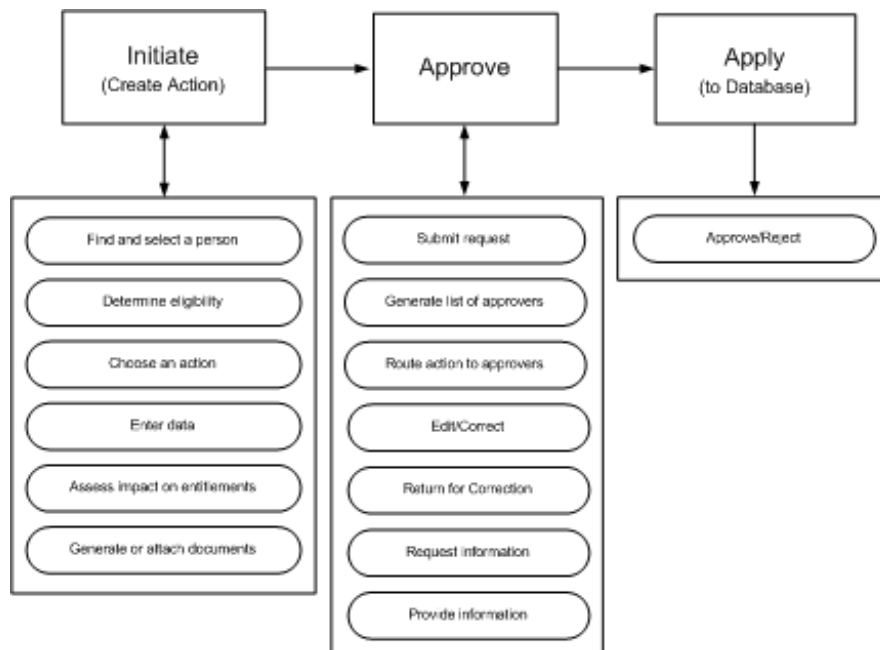


Figure 12 – Oracle SSHR – Action Lifecycle

Further configuration is possible in the following areas:

- Set system profile options;
- Define access roles;
- Personalize pages;
- Add a sub menu to user menus;
- Set up eligibility processing;
- Set up document management.

In conclusion, similarly to HCM P&F, Oracle SSHR provides several configuration tools. Table 2 shows a comparison between these two solutions:

SAP HCM Processes & Forms	Oracle Self-Service Human Resources
Pros <ul style="list-style-type: none"> • Built within SAP ERP (“out of the box” access to all data) • Highly configurable • Custom actions • Custom logic (code) possible 	Pros <ul style="list-style-type: none"> • Built within EBS (“out of the box” access to all data) • Highly configurable • Customizable Page Layout • Custom logic (code) possible
Cons <ul style="list-style-type: none"> • Limited UI design supported by SAP (Adobe, Web Dynpro) • Performance 	Cons <ul style="list-style-type: none"> • No custom actions • Oracle implementations frequently take more time than SAP’s (Panorama Consulting, 2015, pp. 9, 12)

Table 2 – SAP HCM P&F vs. Oracle SSHR

2.4.2 SuccessFactors Employee Central

SuccessFactors⁸ (SFSF) is a multinational corporation, founded in 2001 by Lars Dalgaard, and known for providing cloud-based HCM solutions using Software as a Service (SaaS) model (SuccessFactors, an SAP Company, 2016). SuccessFactors was acquired by SAP in a process that started on December 1st, 2011 and ended on February 16th, 2012. It was also announced that SuccessFactors would remain independent and named “SuccessFactors, an SAP Company” (SAP SE, 2011).

SuccessFactor’s primary product is the SaaS with the same name. SuccessFactors (product) provides several HCM solutions: **Employee Central**, Talent Management (Recruiting, Mobile, Learning, Performance & Goals, Compensation and Succession & Development), Analytics and a social collaboration platform called SAP JAM.

Employee Central is SuccessFactor’s core HR solution to manage the various employee events and event reasons that can occur during the lifecycle of an employee (new hires, transfers, promotion, etc.) (Marson, et al., 2016, p. 107).

The organizational structure in Success Factors is very similar to SAP’s (Mistry, 2015):

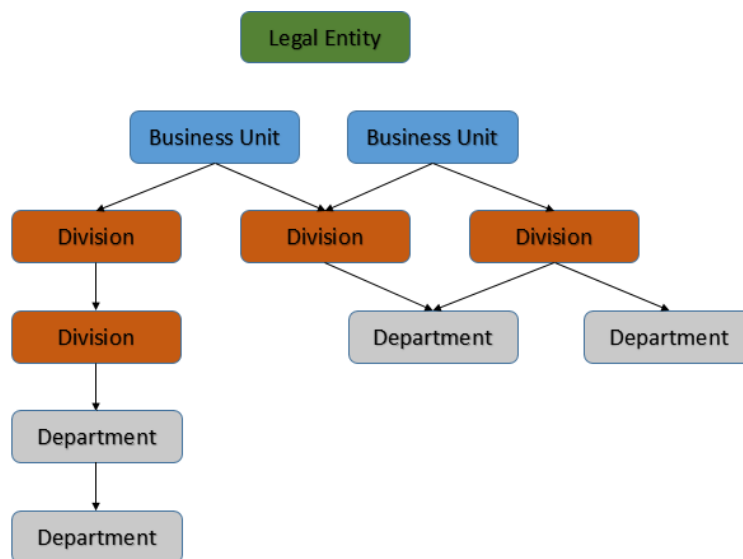


Figure 13 – SuccessFactors – Organizational Structure

Users execute **HR processes/actions** from the website. As an example, Figure 14 shows the first step of a new hire process started in SuccessFactors.

⁸ http://www.successfactors.com/en_us.html

The screenshot shows the 'Add New Employee' form in SuccessFactors Employee Central. The form is titled 'Add New Employee' and has a 'Identity' section. The fields are as follows:

- Hire Date:** Feb 1, 2016
- Company:** No Selection
- Event Reason:** New Hire (HIRNEW)
- Name Information:**
 - First Name:** Fábio
 - Last Name:** Ramalho

Figure 14 – SuccessFactors Employee Central – Hire Employee

Every HR action is configurable within SuccessFactors. It's possible to show/hide fields and sections. This can be done through an XML file called Succession Data Model (SAP SE or an SAP affiliate company, 2015, p. 91) or administration pages within SuccessFactors itself (SAP SE or an SAP affiliate company, 2015, p. 102). The flow of HR actions is controlled by Approval Workflows (SAP SE or an SAP affiliate company, 2015, p. 258). Regarding workflows, it is possible to set up: approval steps, contributors and cc-roles (SAP SE or an SAP affiliate company, 2015, p. 259). Further customization is possible with MDF, a framework that allows companies to create custom objects (fields, tables, etc.), business rules and add them to the UI (SAP SE or an SAP affiliate company, 2015, p. 7).

Considering this, it's fair to say that the goals of SuccessFactors Employee Central are very similar to the ones achieved by HCM Processes & Forms. However, there are some differences in the implementation processes:

SAP HCM Processes & Forms	SuccessFactors Employee Central
Pros <ul style="list-style-type: none"> Built within SAP ERP ("out of the box" access to all data) Highly configurable Custom actions Custom logic (code) possible 	Pros <ul style="list-style-type: none"> Cloud-based Easy and intuitive configuration (MDF)
Cons <ul style="list-style-type: none"> Limited UI design supported by SAP (Adobe, Web Dynpro) Performance 	Cons <ul style="list-style-type: none"> Limited customization Custom logic (code) not possible No custom actions in EC For companies still using SAP ERP, all data must be migrated (or integrated) – an expensive task

Table 3 – SAP HCM P&F vs. SFSF EC

2.5 Assumptions & Restrictions

Before starting the process of finding a solution, there are some restrictions that need to be considered. As described earlier, this solution is directed at a customer who has all of its HR data in SAP ERP. This means that the preference will, obviously, go to a solution supported by SAP. Otherwise, we need to replicate data into an external system, task that would increase hugely the cost of implementation.

Other circumstance that points to an SAP solution is the fact that the team working on this project has more than 3 years of experience in SAP and specifically, have been working with HCM Processes & Forms.

However, as stated in a previous section, this company already has an HR application – for readability, this application is from here on called as Personnel Change App. Despite all the problems previously identified, this application has a good response to user action – performance is not an issue. Therefore, the performance characteristic must be the same in the new application. This means that most synchronous user actions (like selecting the value of a dropdown and waiting for other fields to be filled) should not exceed 2 seconds.

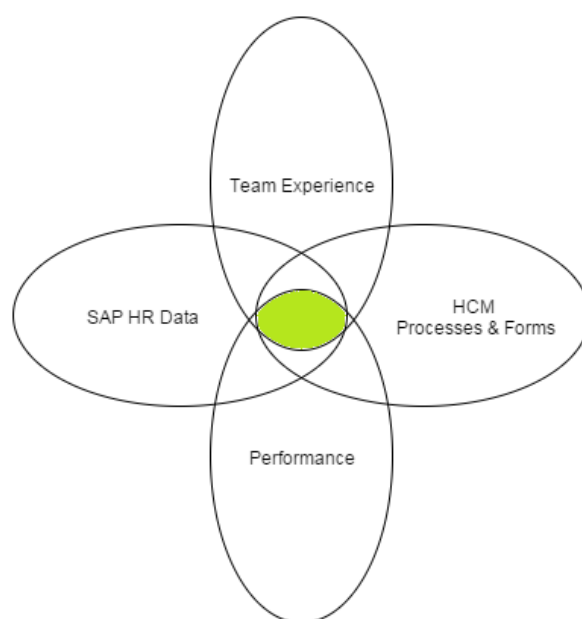


Figure 15 – Restrictions

Besides these restrictions, it is assumed that:

1. SAP ERP's version is 6.0, with at least Enhancement Package 2 (SAP SE, 2016k).
2. Personnel Administration module is setup.
3. SAP HR Data is up to date and coherent with the company standards.

2.6 Value Analysis

For any business scenario, a good value proposition (VP) is crucial to reach out to possible customers and let them know the value of your products/services. It “is an overall view of a company's bundle of products and services that are of value to the customer” (Osterwalder, 2004). In other words, it may determine someone’s interest in your products and distinguish you from the competition. If you do not have a good VP, chances are you’re losing potential customers who don’t understand the value of your products to them. To establish a good VP, it is important to know how to define and measure value. Value is associated with the importance, benefits and usefulness of your product. Understandably, the value perceived by the customer is different due to the fact that customers consider costs they will have (besides the benefits). This is the primary reason for VPs – they are the best way to influence customer’s perceptions.

The product described in this dissertation is an application that allows company users to manage, execute or approve HR actions, leveraging and extending SAP ERP’s functionalities. Figure 16 shows the benefits and sacrifices (from pre-purchase phase to disposition phase) for customers who buy this product.

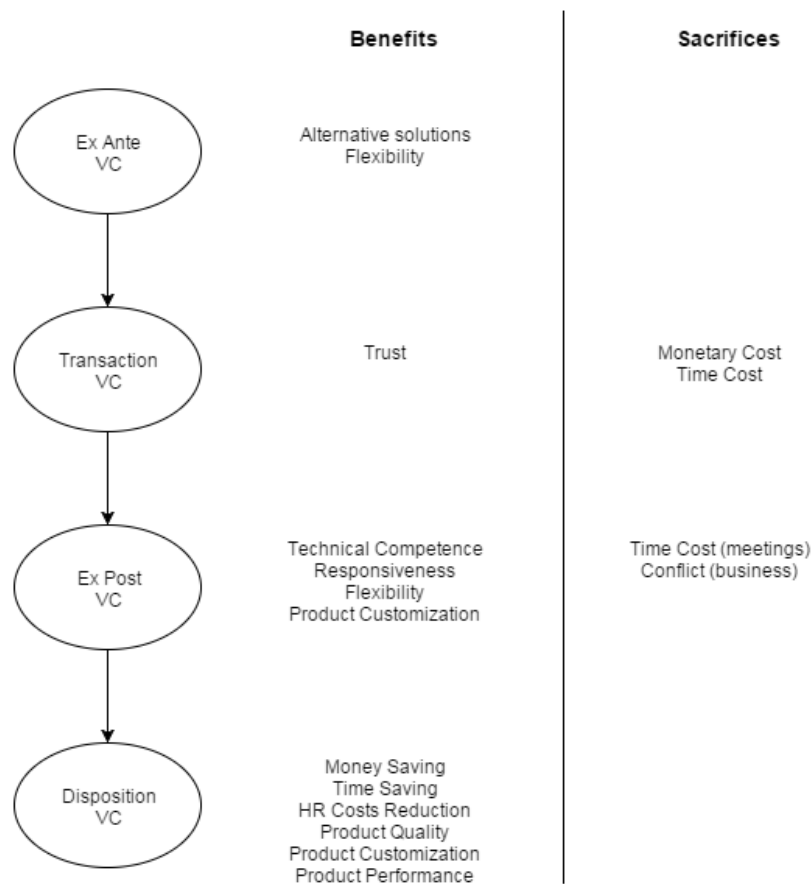


Figure 16 – Value Analysis – Benefits vs. Sacrifices

The application is directed to the Oil & Gas industry. However, it is suitable for any enterprise provided that the customer uses SAP ERP as their HR system of record. Since the product is easily customizable, it can be adjusted to the specifications of any customer. For further details on the business model here presented, please check the Business Model Canvas under Appendix 1 – Business Model Canvas.

To analyze the value of this product, Analytic Hierarchy Process (AHP) was used, a structured technique for dealing with complex decisions. Through AHP, it was possible to compare the value of this solution to the ones described under section 2.4 State of the Art in Solutions (considering the pros/cons already described). To help quantify the different alternatives/products, a report called *Clash of the Titans 2016: An Independent Comparison of SAP, Oracle and Microsoft Dynamics* by Panorama Consulting was analysed. This report contains essential information on the statistics of implementing ERP projects.

Four factors were used in this analysis: performance, configurability/customization, implementation costs and reliability. The following picture summarizes the results per factor.

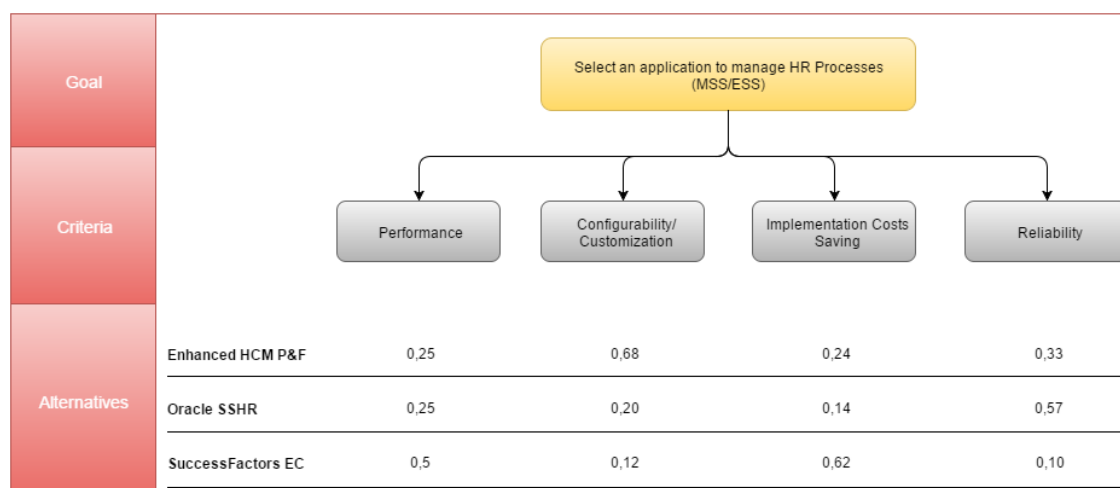


Figure 17 – Value Analysis – AHP Diagram

These results tell us that the solution preconized in this document surpasses the other two in terms of configurability or customization, but it lacks the same reliability as an Oracle solution, which has been in the market for several years.

To fully understand these results, we must combine the alternatives with criteria weights. This obviously can differ by customer. Some may opt for a low-cost implementation while others favor an application with good response time, even if it takes longer to implement. The picture below shows a possible combination of criteria weights with the alternatives above.

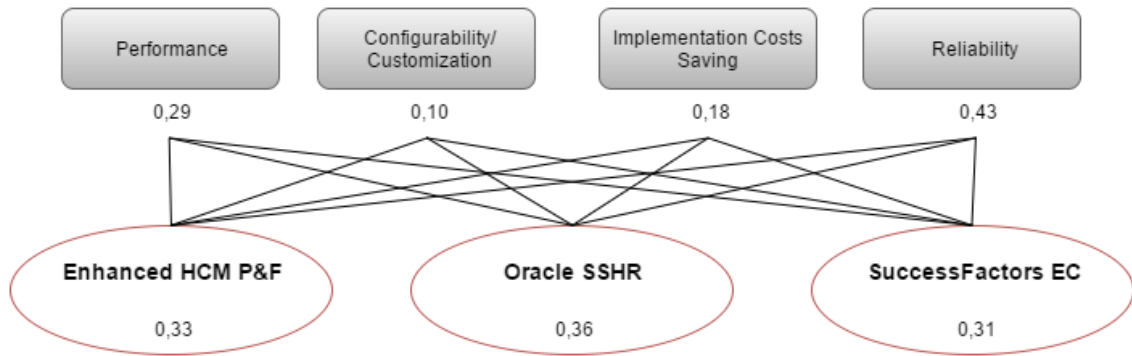


Figure 18 – Value Analysis – AHP Decision Simulation

In conclusion, the absolute value of these solutions is very similar. A customer who strongly values reliability (like in the example above – reliability = 0.43) will be inclined to choose a standard solution, but if we look at the remaining criteria, the solution described in this document is a better option.

To view the complete AHP analysis, please go to Appendix 2 – AHP.

3 Solution Analysis

This chapter contains two approaches under analysis and how they lead to the final solution.

3.1 Personnel Change App (Legacy Application)

As described in section 2.5 Assumptions & Restrictions, the company to whom this product is molded already uses a web application to manage their HR processes. One of the approaches to consider is the possibility of enhancing this application, making it more powerful. In order to measure the viability of this approach, the following metrics were used:

- User Satisfaction
- Support/Maintenance Effort

To evaluate user satisfaction, four groups were established:

- Requesters (employees who use the application to request HR changes)
- Direct Managers and HR Managers
- Compensation/Benefits Managers
- SAP Administrators (employees that assure data is properly saved inside SAP ERP)

Each one of these groups consisted of people at a high level in the organization. Through meetings and conference calls, they were able to compile a list of the employees' ideas, complaints, wishes, etc.

Requesters were very critical in regards to the application, being most of the reasons:

- The amount of data they had to fill, without any hints from the application.
- The time they spent filling that data (users reported that sometimes they would spend a full hour handling just one request).
- They also reported that several errors occurred and they couldn't understand the reason, because messages were too technical.

On the other hand, the managers' groups transmitted that they were satisfied with the application. They were able to rapidly approve requests and continue their work. So, they didn't expect major changes in the application business flow.

SAP Administrators reported that the application was not aware of the business rules in place. Most of the times, data entered by requesters and managers could not be posted to SAP at the last stage of application flow. For example, if a requester is doing an international transfer for a given employee, but then selects as reason for action in-country merge, that data would not be accepted by SAP. Besides, several times SAP Administrators would get errors due to data

inconsistency in SAP HR database. Although this is not related to the web application, the issue here is that SAP Administrators are only informed of this error once they try to process the request within SAP ERP. This leaves no choice but to manually update data and validate consistency (a task that makes the application useless). The ideal scenario would be to inform SAP Administrators in advance, so they can solve the data issue.

These facts alone are not sufficient to determine if this application should be reused. However, it's clear that, to fulfill user requests, some modules need to be completely restructured.

In order to evaluate the cost of maintenance/support, a more technical method was used. Personnel Change App is an application with low dynamism, SAP logic is being replicated and most of it is hardcoded. Communication between the web application and SAP ERP is made through web services. For each field with a specific logic, a new web service exists:

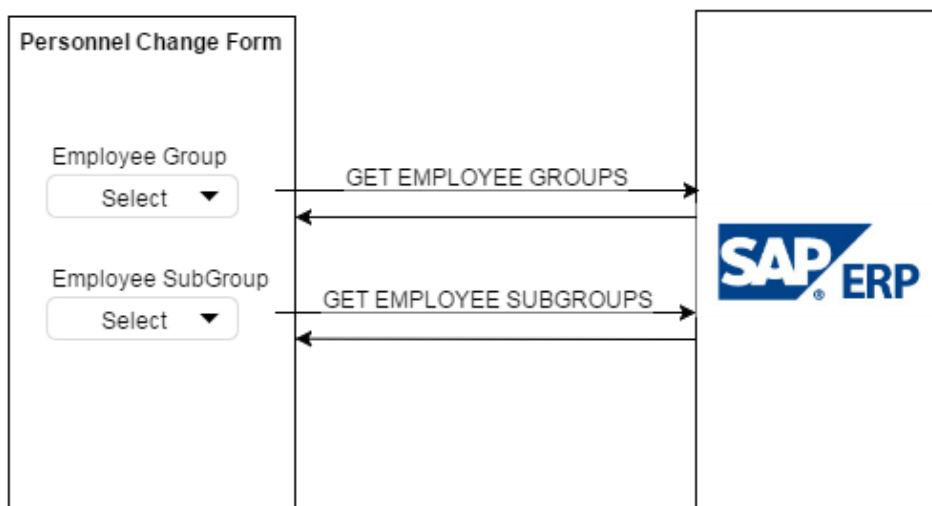


Figure 19 – Communication between Personnel Change App & SAP

Currently, these web services don't take advantage of SAP standard rules (as correctly pointed by SAP Administrators). For this reason, they would have to be completely revamped, a task that would require a tremendous effort. In terms of maintenance, for every new rule added in SAP, we would need to make enhancements to the application, in both backend and front-end. This goes against one of the pillars of the solution – easy configuration and customization. So, the viability of replicating all SAP logic using the current infrastructure is highly questionable.

The average time to implement new field logic in Personnel Change App is 2 days. Figure 20 shows the expected effort of revamping Personnel Change App and continuously add new logic vs. the equivalent effort if using an application that leverages SAP business rules.

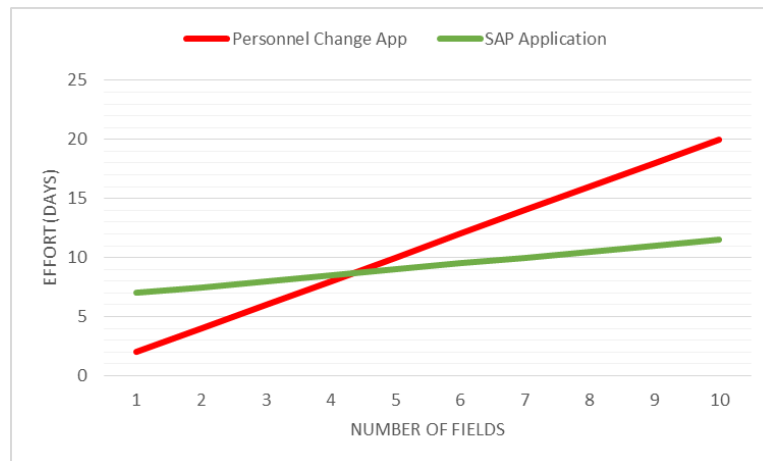


Figure 20 – Personnel Change App vs. SAP Application

Looking at this graphic, it's possible to infer that, using Personnel Change App, the development time is proportional to the number of fields / web services. On the contrary, a new application would require more time to setup, but little or no effort for the subsequent activities.

3.2 Enhanced SAP HCM Processes & Forms

SAP HCM Processes & Forms was presented under section 2.3 State of the Art in Technology. As described in that chapter, this framework is part of SAP Portal's solution and has a very limited UI design. These two characteristics are prohibitive for the customer, so we can already say that a full standard HCM P&F implementation is not a viable solution.

Additionally, to validate this framework's utility, the following characteristics were used:

- Flexibility (to support any front-end technology)
- Performance / Response Time

To measure the flexibility of this framework, an extensive analysis was made to the standard SAP code, specifically the access points and communication between all the components. It was discovered that the entry point is a function module⁹ called *ISR_PROCESS_EVENT*. This function is used in all HCM P&F functionalities. It is invoked when a process starts, events are triggered, data is saved, approved, rejected, etc. A detailed analysis on this function's signature is available in Appendix 3 – SAP HCM P&F Analysis.

With that in mind, the next step was to develop a proof of concept to show that this function can be used to replicate SAP Portal's behavior. This POC consisted of a simple HR process with minimal amount of form fields and one-step workflow. In addition, a program was created to simulate the action of starting a process (using the ISR function). The test was a success, fact

⁹ Function Modules are procedures written in ABAP, SAP's primary programming language.

that proves the ability to combine HCM P&F with any presentation layer technology. For instance, the ISR function can be easily wrapped in a Web Service.

In order to evaluate performance, the same POC was used. A few more fields were added to the form so that the amount of data would be big enough to take conclusions. In the first tests, the creation of a process would take up to 50 seconds. These results are intolerable for end users, so a runtime analysis was made.

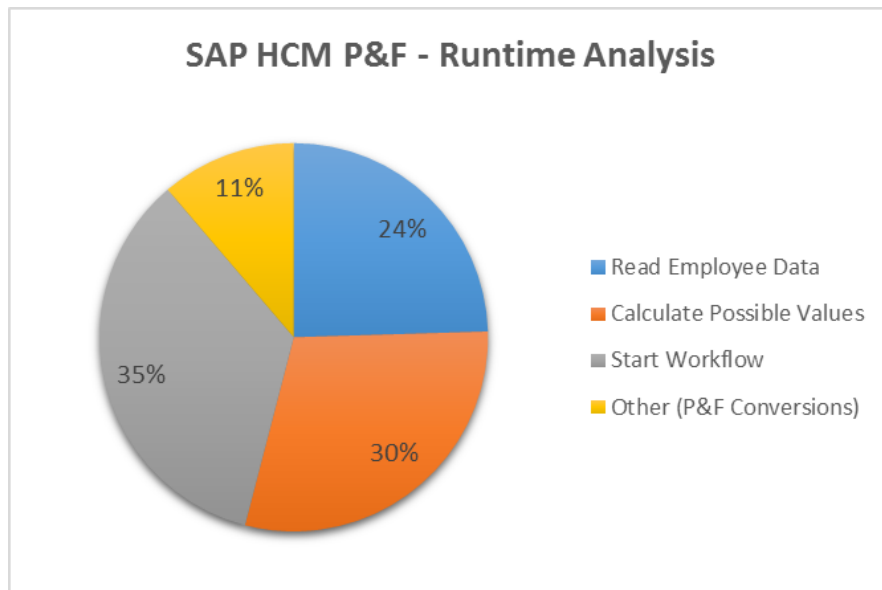


Figure 21 – SAP HCM P&F – Runtime Analysis

“Start Workflow” is the last step in process creation. This step is asynchronous, which means that performance can be improved if some of the other steps are shifted. From the analysis of *ISR_PROCESS_EVENT*, it was found that a parameter can be used to exclude the calculation of possible values at process creation (in SAP Portal, this is meant for creating new processes in mass). So, this parameter was turned on and the calculation of possible values was triggered directly by the program (by calling specific HCM P&F functions), while the workflow was being started.

In order to analyze the performance improvement, a paired-sample t test was used. In average, the execution time dropped 13,58 seconds – this corresponds to 36% of the total execution. Detailed information can be found in Appendix 3 – SAP HCM P&F Analysis (2nd part).

There are also other performance tuning points in HCM P&F (Prabhu Mithal, 2010), as stated by several users in the SAP community. To summarize, it’s possible to work around HCM P&F performance.

3.2.1 Other Projects in SAP HCM P&F

Projects involving HR Processes are usually developed internally or with the help of external companies that assure experience and confidentiality. This is also the case with HCM P&F implementations.

Since these projects are normally confidential, none or little public information exists that would help the process of evaluating this framework. Nonetheless, an article regarding one of these implementation, wrote by Christopher Solomon, a recognized HCM P&F consultant, was reviewed during this process. This article is called *HCM Processes & Forms: Gotchas, Bugs and Other Curiosities*¹⁰ and takes the reader through some of the limitations faced during an implementation of HCM P&F.

The first thing to take from this article is the reference to PCR (Personnel Change Request), a legacy framework that was decommissioned a few years ago with the rise of HCM P&F. The writer acknowledges the improvement brought by HCM P&F: “That being said in my opinion, HCM P&F removes a lot of the limitations PCRs had and also adds loads of functionality. I will start off by saying, I think it is a great framework!” (Solomon, 2008). However, the writer also refers that by the time this framework was launched, it contained several problems. For that reason, the advice is to use HCM P&F from Enhancement Package 2 on: “Even on our project, our beginning days with the delivered HCM P&F were trying to say the least. Once we upgraded to Enhancement Pack 2 for EA-HR, not only were many of our bugs/errors/etc corrected, but it also added even more functionality to HCM P&F. To that point, I would highly suggest that you at least start from EhP 2 when working with HCM P&F” (Solomon, 2008). This suggestion was obviously considered and defined as restriction, as previously presented under 2.5 Assumptions & Restrictions.

After this, the writer enumerates some HCM P&F bugs and helpful hints to tackle them. Some of these issues are irrelevant to our implementation or were, in the meantime, corrected by SAP SE, so, just the necessary ones will be addressed.

Rules must use fields from service

The writer acknowledges the good feature that rules are: “A nice feature of HCM Process and Forms is that you can define “rules” and then use these “rules” to determine if an entire service will trigger or even if operations within the services will occur.” (Solomon, 2008). The issue here is that one cannot setup a Back-end Service and then create a rule based on form fields that are not assigned to that Back-end Service. “For example, we have some radio buttons on a form that might determine if the user gets bonus pay or not, so we have a rule to check “IS_BONUS = ‘YES’”. However, when doing an infotype operation from standard service SAP_PA and wanting to use that rule to decide if the user gets the bonus or not, of course, the standard infotype does not have our little “is_bonus” field in it.” (Solomon, 2008). There is no simple workaround, as stated in the article, so it forces us to rethink the way rules are setup (for

¹⁰ <https://blogs.sap.com/2008/06/30/hcm-processes-forms-gotchas-bugs-and-other-curiosities/>

instance, decide if Back-end Service should run based on the fact that fields assigned to it were changed). The worst scenario would be the need to create a custom Generic Service (Solomon, 2008).

Multiple operations not allowed for an infotype

This limitation means that HCM P&F are not prepared to handle multiple operations for the same infotype. For example, having the possibility of creating new records and modifying others without delimiting them is not supported. There is no workaround apart from developing a custom Generic Service (Solomon, 2008).

Only 3 field values can be read from Form into Workflow

This is not a critical limitation, but still worth mentioning. SAP SE provides a way of fetching data from the form into the workflow for subsequent decisions. However, only 3 fields can be read at a time (this is a limitation in the background task used for this purpose). There are a few workarounds: one would be to call this task several times in the workflow, but the more efficient way would be to develop our own task and read the amount of data necessary in one call (Solomon, 2008).

User Events

Here, the writer alerts to user events and explains how they work as simple triggers for the operations attached to form fields. User events should be handled carefully and extensively tested (Solomon, 2008).

At the end of the article, despite the negative aspects pointed out, the writer concludes there is a big potential in this framework and remembers that the purpose of the article was to “make people aware of the obstacles they might face and how to deal with them rather than getting turned off by HCM P&F all together” (Solomon, 2008).

Considering the extensive analysis detailed in these last sections, we are now aware of almost all the existing limitations and how to work around them. This gives the confidence necessary to consider an HCM P&F implementation (with proper enhancements) a viable solution.

4 Design

Considering the solutions presented earlier and the fact that SAP HCM P&F is the preferred approach, this chapter contains the overall architecture of the preconized solution, including analysis of requirements (functional and non-functional) through the use of UML diagrams.

4.1 Functional requirements

4.1.1 Actors

There are six main actors in the application: Requester, HR Manager, Direct Manager, Compensations/Benefits Manager, SAP Administrator and Payroll Administrator.

Requester is the person that proposes data changes for someone in the organization, by creating a ticket and filling form data (i.e. starts a new HR process). Depending on authorizations, they can create a ticket for themselves.

HR Manager and **Direct Manager** are responsible for approving the ticket.

Compensations/Benefits Manager is responsible for validating and editing data related to employee's salary, bonus, etc.

SAP Administrator is a technical employee with SAP knowledge who will handle the process of persisting changes to the system.

Payroll Administrator is responsible for reviewing data and acknowledging it.

4.1.2 Ticket Types

The application should allow the creation of the following personnel change requests, known as ticket types:

- Hire Employee/Contractor
- Re-Hire Employee/Contractor
- Contractor to employee
- In-Country transfer/International transfer
- Termination/Suspension
- Address Change & Self Address Change
- Change Cost Center, Manager, Work location

4.1.3 States

The solution here presented focuses on the application backend, specifically the modules tied together with HCM P&F. To allow further customization on the web application, this solution should be prepared to map several states on the web application to one in SAP. For example, in SAP there should be a generic state (Form Edition), but in the web application it can be divided in several sub-states (Requester Form Edition, Employee Form Edition, etc.). Furthermore, although there are several ticket types, the business flow is similar to all of them, so a generic workflow will be built. The following diagram shows the states relevant to SAP:

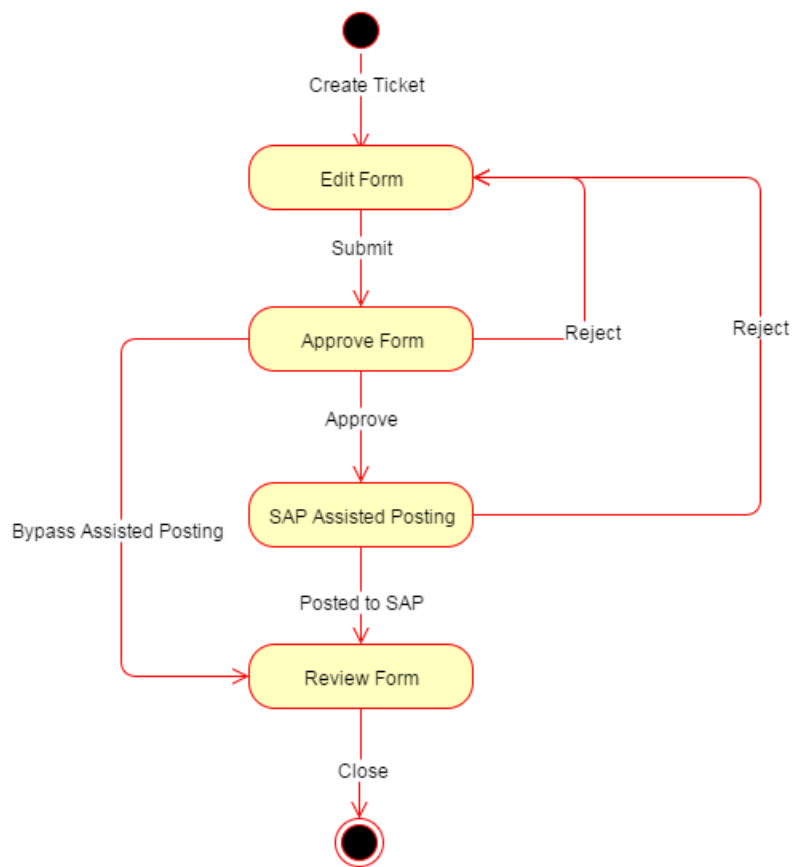


Figure 22 – State Diagram

The first stage – Edit Form – results of the ticket creation. After selecting the target employee and ticket type, the user is responsible for filling form data and submitting it for approval.

Afterwards, the ticket goes into the approval stage, where managers are responsible for approving/rejecting the changes. Once approved, the ticket moves into SAP.

In SAP Assisted Posting stage, SAP Administrators are responsible for processing the ticket data from their inbox within SAP ERP – SAP Business Workplace¹¹. Here, SAP Administrators will have automated functions to reduce their manual effort.

There should also be a last stage to allow SAP Administrators and Payroll Administrators to review the form and apply manual changes if necessary.

4.1.4 Use Cases

The following diagram shows a simplified version of the actions each user can take:

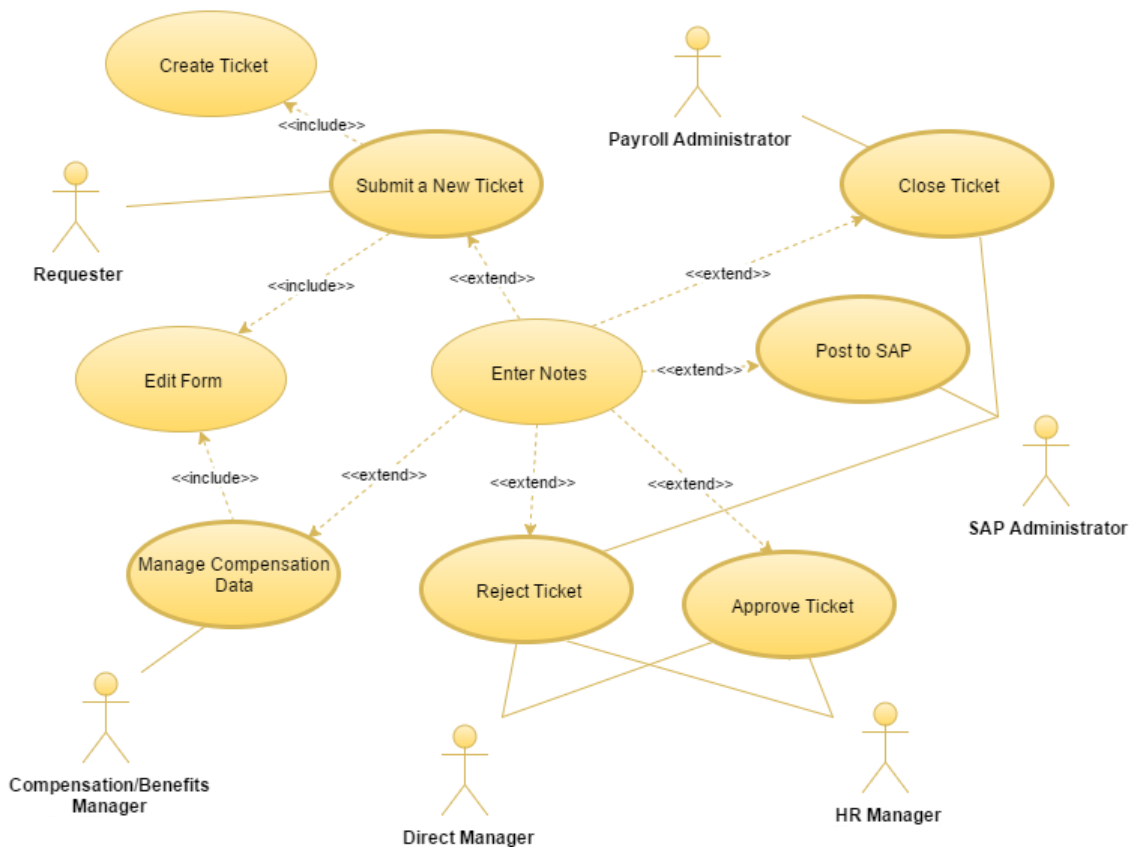


Figure 23 – Use Cases Diagram

For detailed information on the primary use cases, please see Appendix 4 – Use Cases.

4.1.5 SAP HCM P&F Adaptation

Requirements presented in the previous sections need to be adapted to HCM P&F framework. Considering the lessons learned in chapter 3 Solution Analysis, this is how it needs to be setup:

¹¹ https://help.sap.com/saphelp_nwpi71/helpdata/en/ae/b82cc8e10611d2a62f0060087a79ea/content.htm

1. Normally, each process in HCM P&F corresponds to one HR action (i.e., one process for hires, another one for promotions, etc.). However, our goal is to create a generic solution for all HR business processes. Besides, a lot of these HR actions share the same fields. For these two reasons, only one process will be setup.
2. All form fields must be added to the Form Scenario, including the mapping to HR infotypes and default/possible values.
3. For each business step, a scenario stage will be created under the Form Scenario.
4. These stages will be reasonably mirrored in the SAP Workflow.
5. User events will be created for every action triggered by user action.
6. Generic Services will be created to hold form-specific logic.

For further details on the HCM P&F architecture, please see 4.3.2 SAP HCM Processes & Forms under 4.3 Overall Architecture.

4.2 Non-functional requirements

4.2.1 Performance

One of the flaws of HCM P&F is its performance. This framework handles a great amount of data and can undermine user experience. For that reason, all the access points in the framework must be used to improve performance. In addition, all the custom code must be efficient and prepared to handle big requests.

4.2.2 Scalability

HCM P&F allows us to develop a scalable application. Since most of its components are configuration-oriented, this solution can easily grow on customer demand. For example, if new form fields are added through configuration, the framework processes will adjust automatically.

4.2.3 Maintainability

The solution should be easy to maintain. All modules should be decoupled so that enhancements/fixes are applied in a short period of time.

4.3 Overall Architecture

The solution's architecture takes advantage of two powerful SAP tools: HCM P&F and Workflow. An overview of this architecture can be seen below.

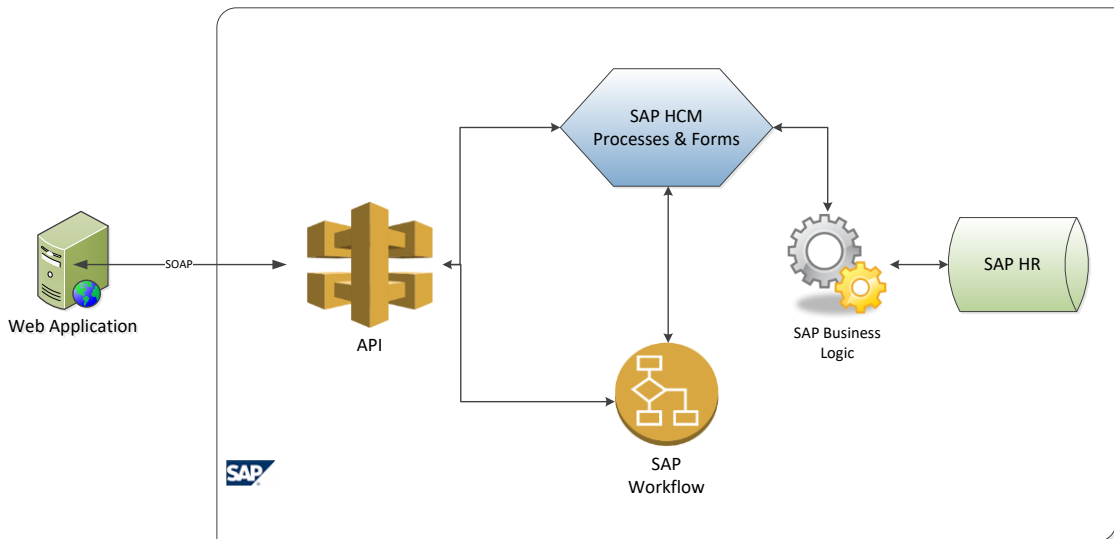


Figure 24 – Solution Architecture

4.3.1 API

The integration API is a communication hub between the web application and SAP ERP. It will handle information related to configuration/design and runtime data. This component is crucial to allow an integration with any front-end technology (remember that SAP only supports Adobe and Web Dynpro forms). This logical layer is built on top of the HCM P&F standard API and will be responsible for handling any event triggered by the web application.

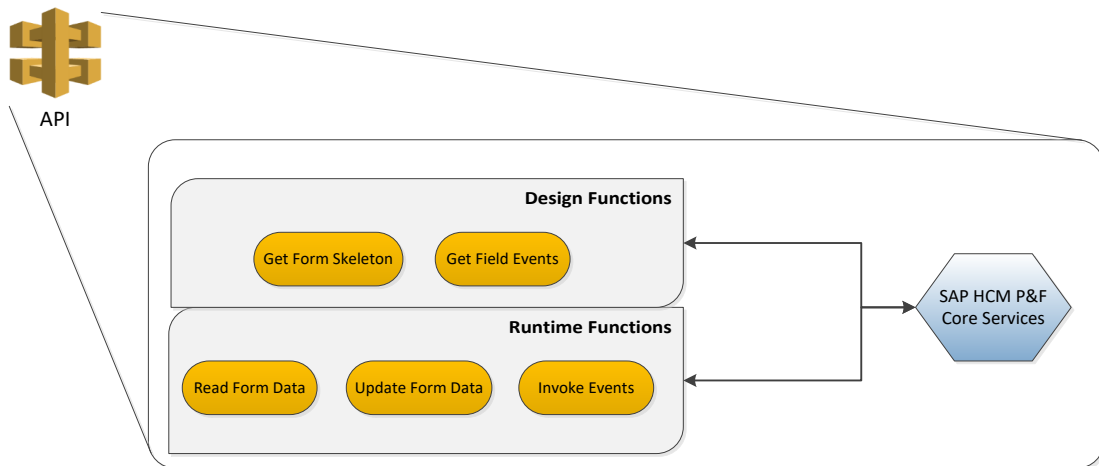


Figure 25 – API Overview

4.3.2 SAP HCM Processes & Forms

SAP HCM Processes & Forms is the centralized source of form fields, along with their SAP behavior. Figure 26 shows an overview of the solution’s HCM P&F module.

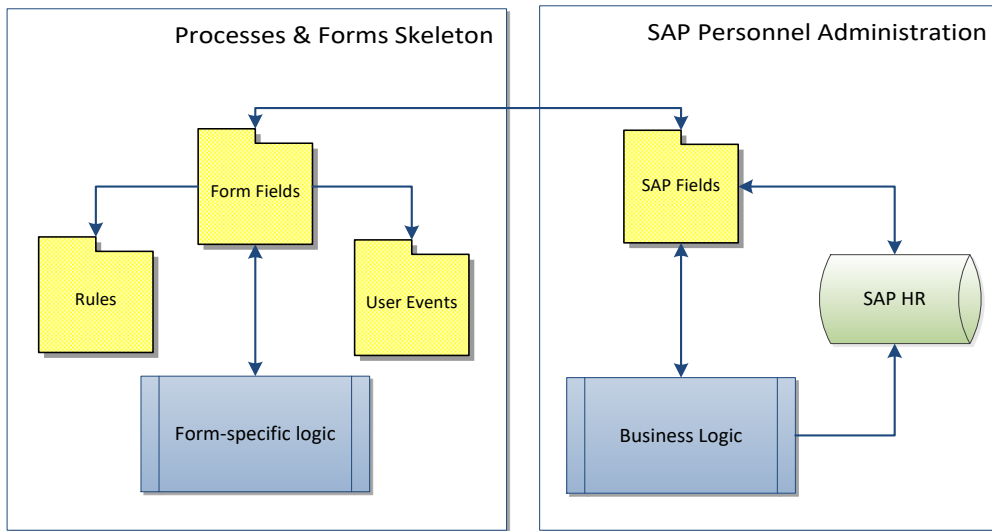


Figure 26 – HCM Processes & Forms Overview

4.3.3 SAP Workflow

SAP Workflow's module is used to specify interactive/background steps that form the generic business process. Figure 27 shows the overall flow of what is desired:

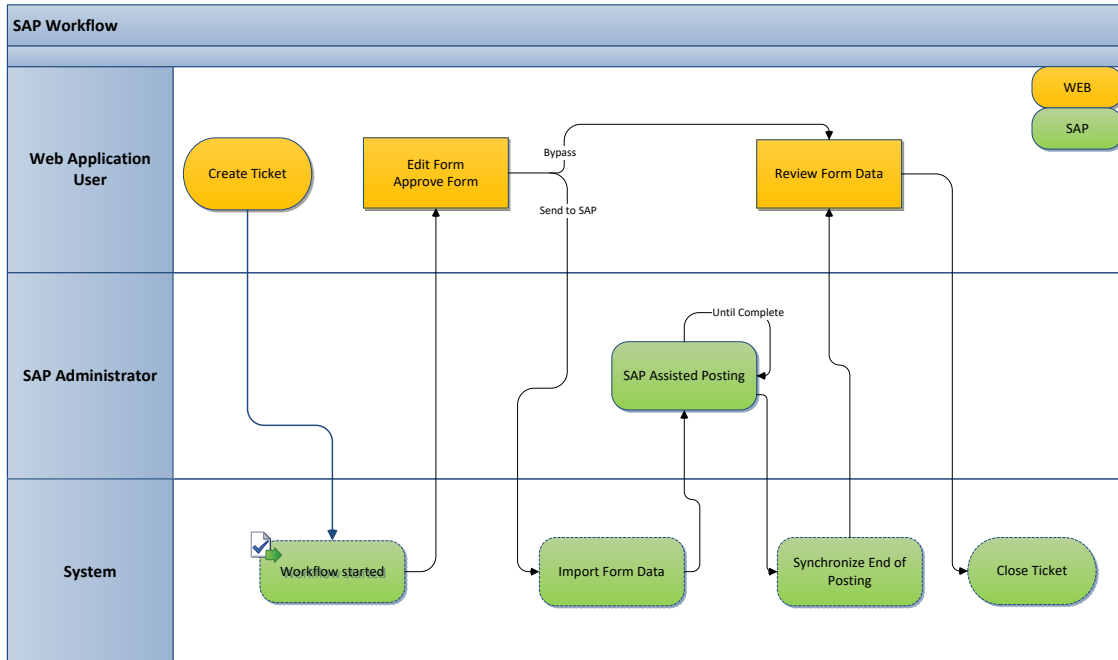


Figure 27 – SAP Workflow Overview

4.3.4 Database

4.3.4.1 Design Time

All configuration done at design time (process, fields, events, etc.) is saved under a vast SAP HCM P&F data model. For the purpose of this project, only a part of this data model is relevant. Furthermore, a few additions must be made to support specific requirements. Figure 28 shows HCM P&F design data model with added value.

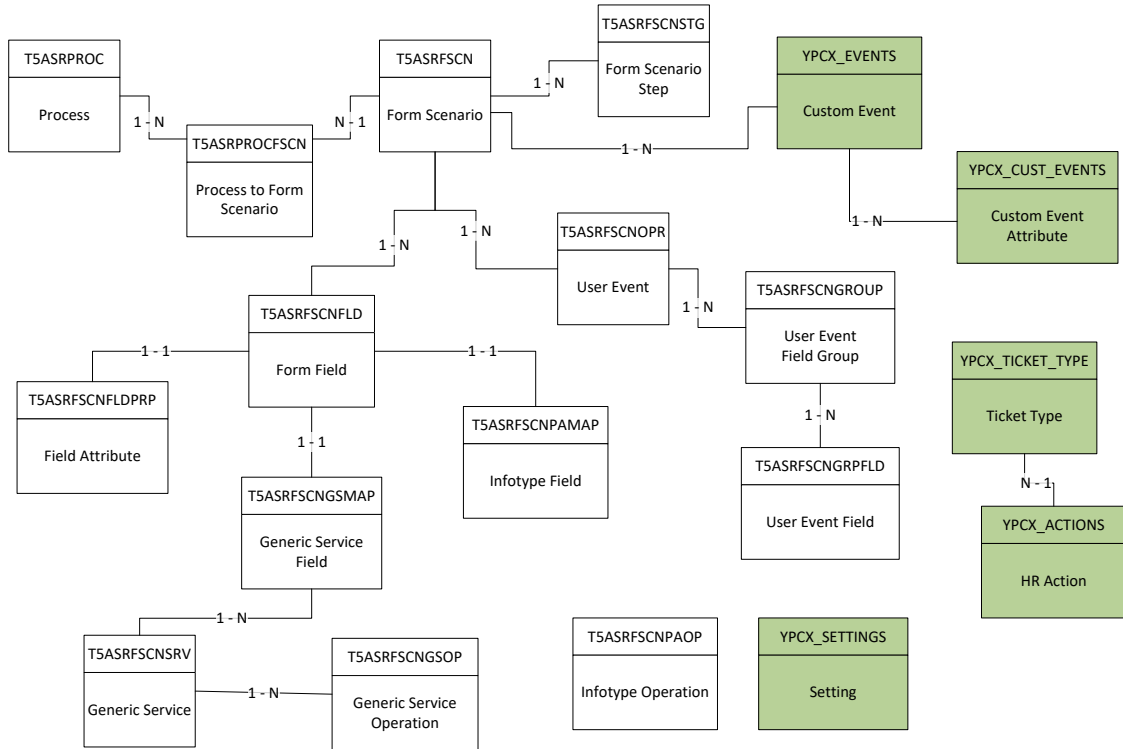


Figure 28 – Design Time Data Model

The purpose of entities Custom Event and Custom Event Attribute is to facilitate the execution of custom logic without going through HCM P&F framework functions. These two tables will hold custom events names, as well as the name of the class/method/function to execute. This structure will help maintain the same dynamism of HCM P&F.

Ticket Type and HR Action hold the mapping between existing ticket types (shown in section 4.1.2 Ticket Types) and SAP HR actions.

Additionally, a settings table will be created to hold functions that can be switched on/off. For instance, performance logs, super users, etc.

4.3.4.2 Runtime

In SAP, data that results from the execution of form-based processes is saved in the Process Object, the persistence layer of HCM P&F framework (SAP SE, 2016r). For each step in the process, an XML file containing form data is created under the Process Object.

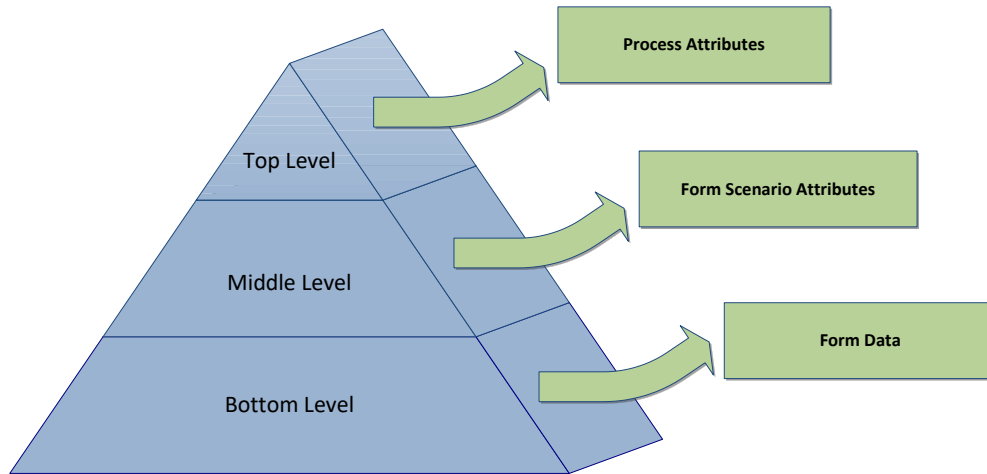


Figure 29 – Process Object Architecture

Remaining process-level data is saved in a simple data model that links processes and workflows:

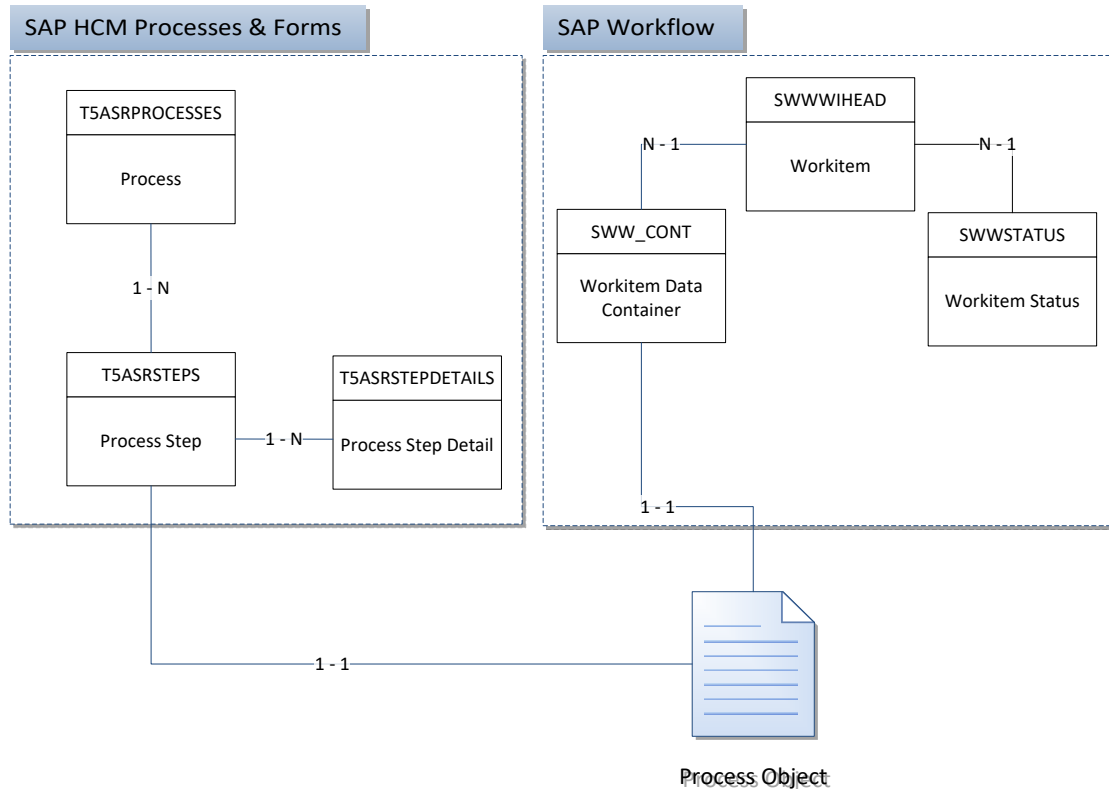


Figure 30 – Runtime Data Model

5 Implementation

This chapter describes the implementation phase of this product, including all the technical objects that were created, the reasons behind each decision and how they relate to what was previously designed.

5.1 Overview

This implementation is subdivided in four big modules, that will be described throughout this chapter:

- API
- SAP HCM Processes & Forms
- SAP Workflow
- P&F External Events

5.2 API

5.2.1 Technical Overview

As previously explained, the API is a bridge between the web application and SAP, built on top of SAP HCM P&F, with the purpose of exposing its features/functions.

Technically, this API consists of:

- An ABAP Class (*YCL_PCX_SAP_API*) which holds all the logic developed to communicate with SAP HCM P&F and handle events triggered externally (in this case, by the web application).
- Function Modules (configured as RFCs¹²), that allow the web application to interact with SAP. These functions are simply wrappers to the class's logic, with the solely purpose of exposing the API.

¹² A Remote Function Call (RFC) is the call or remote execution of a Function Module in an external system. It allows communication between SAP and non-SAP systems (SAP SE, 2016w).

Figure 31 displays an overview of these components:

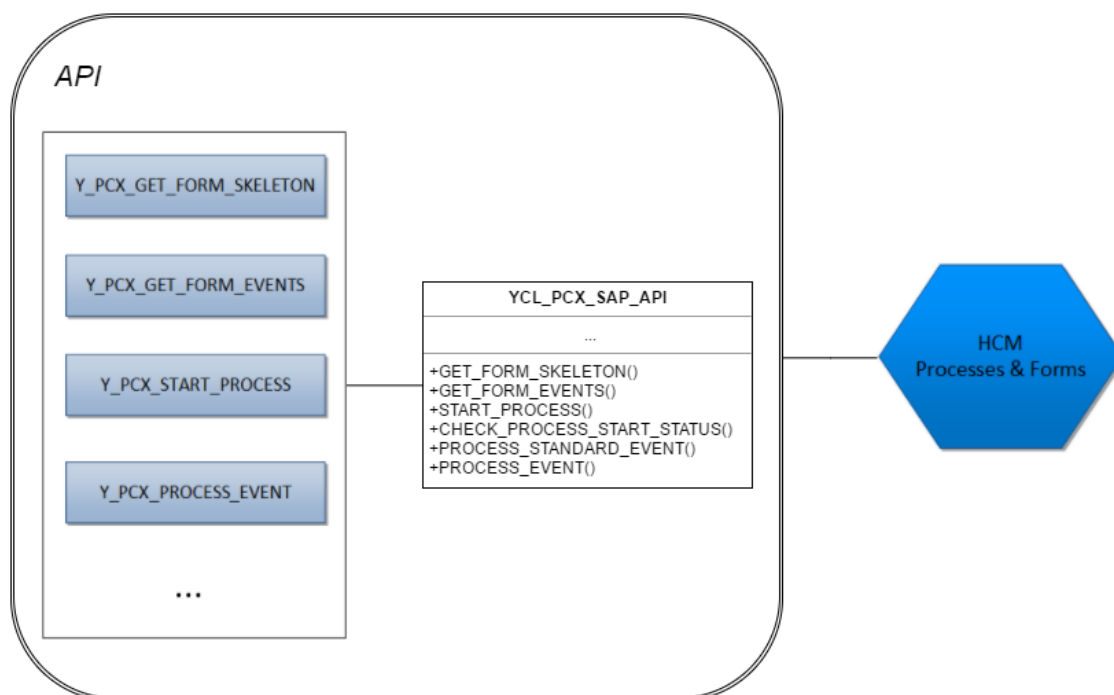


Figure 31 – API objects

5.2.2 Primary functions

This section contains the most important functions exposed by the API.

5.2.2.1 Get Form Skeleton

This function – *Y_PCX_GET_FORM_SKELETON* – is responsible for returning the configuration setup in HCM P&F: **form configuration**, **fields** and **infotypes**. This is a one-time call and it only needs to be refreshed if new features are added to HCM P&F.

The signature contains only one parameter:

Parameter	Type	Data type	Description
ET_FORM_SKELETON	Export	YPCX_FORM_SKELETON_FIELD_T	Table holding the HCM P&F config.

Table 4 – Get Form Skeleton – Signature

Technically, this function module simply delegates the work in class method *GET_FORM_SKELETON*, which reads the HCM P&F standard configuration tables to gather the following information about each field:

- Fieldname, Description, Maximum Length;
- Infotype/Subtype;
- SAP Infotype Fieldname;

- Indicator if the field is part of a multi-record infotype;
- Indicator if the field determines the subtype of the infotype;
- Indicator if the field should be persisted to SAP ERP;
- Indicator if the field is required when performing the full form check.

Figure 32 shows the execution flow of this function:

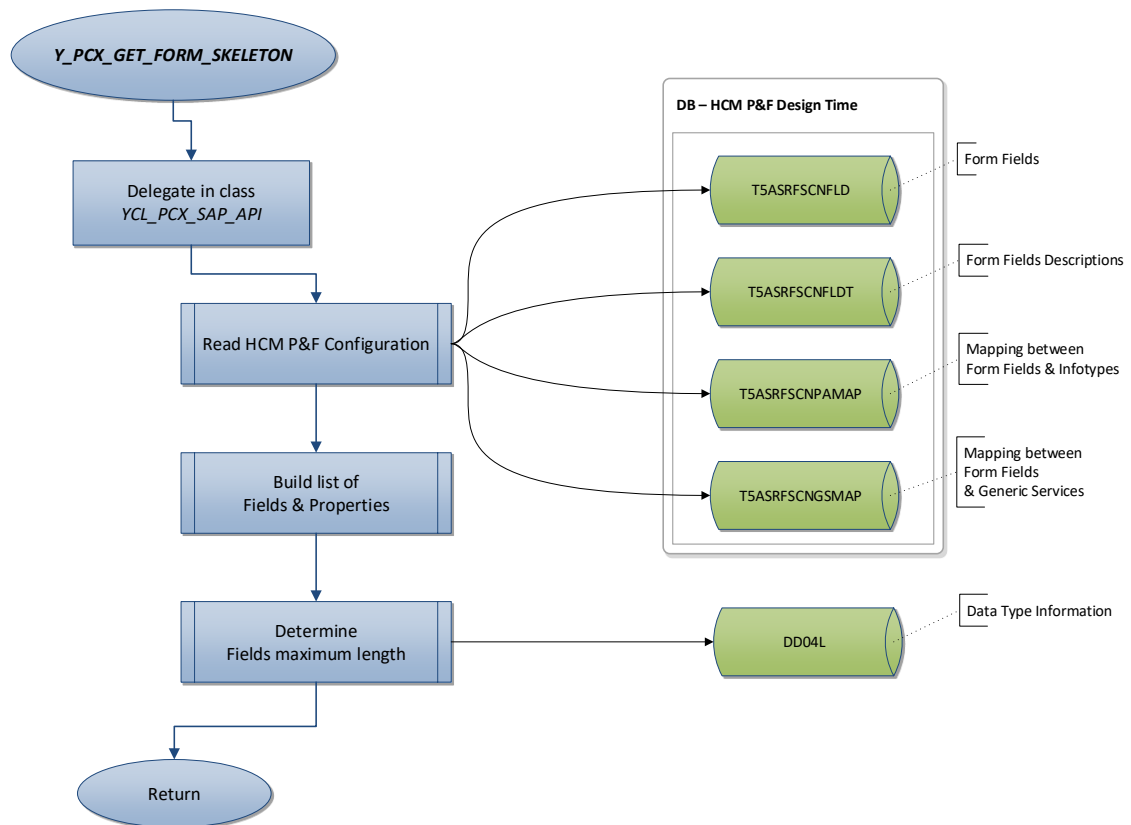


Figure 32 – Get Form Skeleton – Flow of Execution

5.2.2.2 Get Form Events

This function – *Y_PCX_GET_FORM_EVENTS* –, similarly to the one above, is meant to return form configuration. In this case, it is responsible for returning **form events**. This information is cached on the website with the purpose of dynamically knowing when to trigger these events.

This is the function’s signature:

Parameter	Type	Data type	Description
IV_EVENT_NAME	Import	ASR_FORM_OPERATION	Optional. When supplied, gathers information about a specific event
ET_EVENTS	Export	YPCX_FORM_EVENTS_T	Table with list of events
ET_MESSAGES	Export	YPCX_MESSAGE_TAB	Error messages
EV_IS_OK	Export	BOOLE_D	Indicator of successful execution

Table 5 – Get Form Events – Signature

Following the same principle as the other functions, this function module invokes class method *GET_FORM_EVENTS*, which reads the design time database and returns the following information on events:

- Event Name and Type;
- Standard/Custom;
- Indicator for calling after ticket creation;
- Information about the role of each field in the event:
 - Import / Export;
 - Trigger / Not trigger;
 - Mandatory / Not Mandatory.

Figure 33 shows the execution flow of this function:

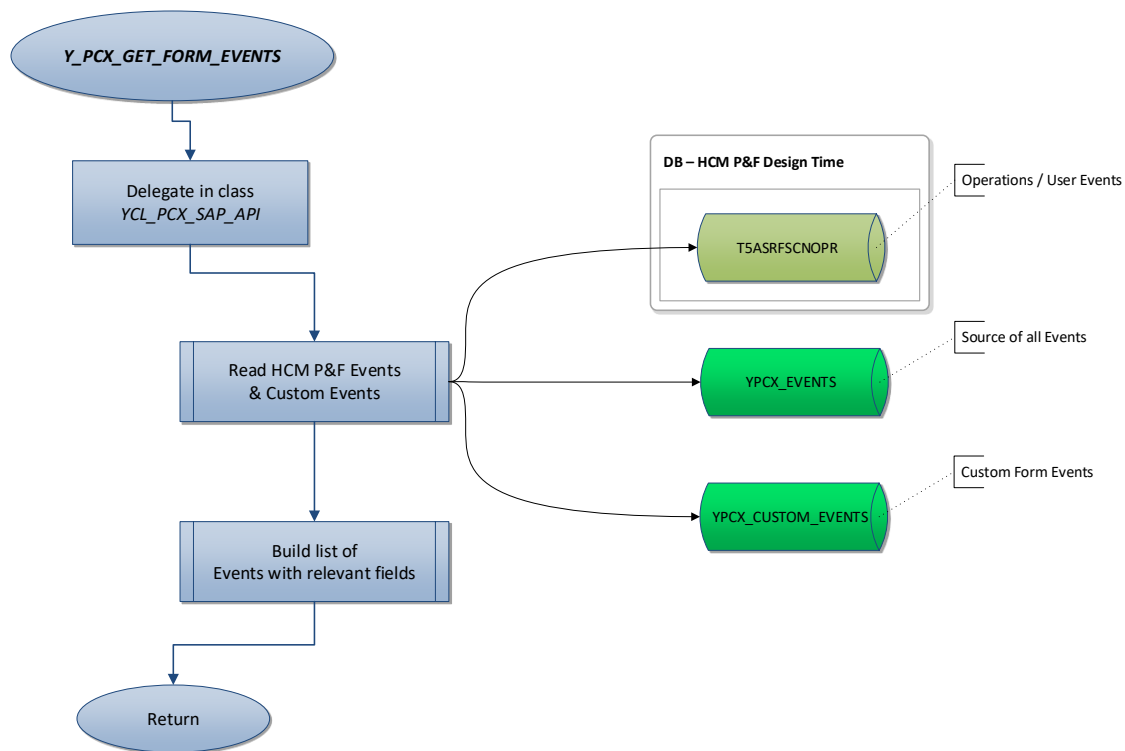


Figure 33 – Get Form Events – Flow of Execution

5.2.2.3 Start Ticket

When a requester chooses to create a new ticket, function *Y_PCX_START_PROCESS* is called to start the process. It requires the following information:

- Ticket Identifier (generated externally);
- Employee ID for whom the ticket is being created;
- Ticket type and Country.

The start of a process in HCM P&F is a lengthy process, perhaps the longest in the framework, even with the tweaks that were applied for this product (as will be detailed later). Additionally, there are events that need to be called right after ticket creation. For these reasons, ticket creation was divided into several phases, explained in Figure 34:

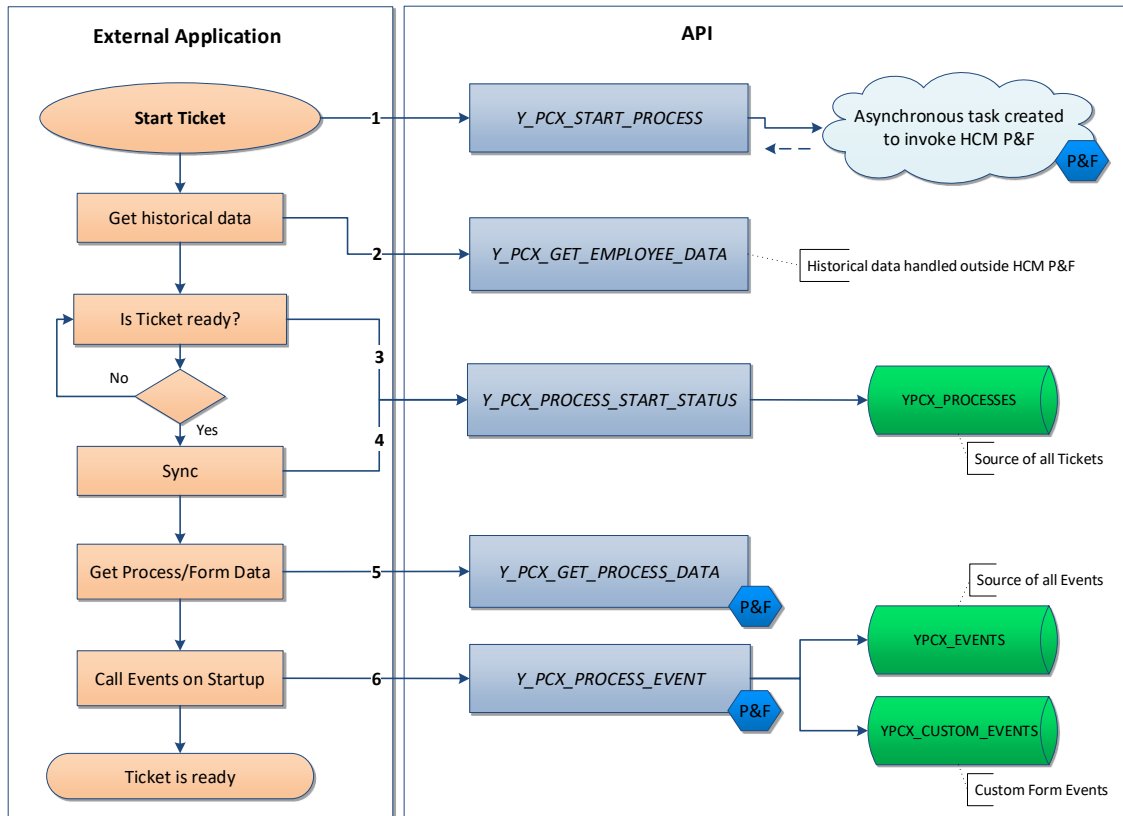


Figure 34 – Start Ticket – Flow of Execution

There are 6 steps to start a ticket:

1. *Y_PCX_START_PROCESS* is invoked in order to create a ticket in SAP. However, as previously stated, this might take some time, so this function module will start an asynchronous call to create the HCM P&F process (through *ISR_PROCESS_EVENT*) and trigger the workflow. The result is the possibility of users starting several tickets in a row or just get back to what they were doing without any block;

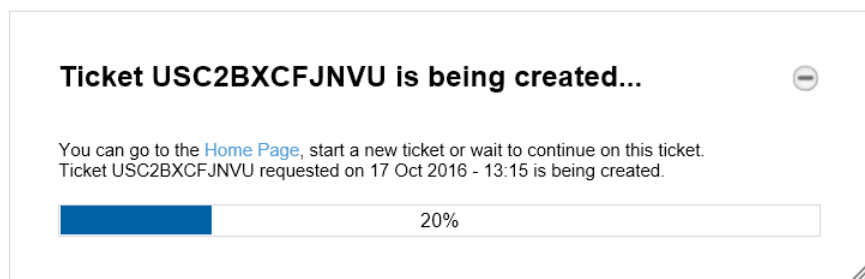


Figure 35 – Web Application – Start Ticket

2. While the ticket is being created in SAP, the web application invokes function `Y_PCX_GET_EMPLOYEE_DATA` to fetch employee historical data or further relevant data that is not part of HCM P&F;
3. At this point, the web application starts contacting SAP, using function `Y_PCX_PROCESS_START_STATUS`, to know if the ticket is ready (i.e. HCM P&F “CREATE” event has ended), according to table `YPCX_PROCESSES`;
4. Once the ticket is ready, the same function is used to acknowledge that the web application and SAP are now synchronized;
5. At this point, the web application fetches all the form data defaulted by HCM P&F, using function `Y_PCX_GET_PROCESS_DATA`. This function accesses the Process Object to obtain this information;
6. Finally, the web application starts calling events that are marked to be executed at ticket creation, using function `Y_PCX_PROCESS_EVENT`, which will be addressed in section 5.2.2.4 Execute events.

After step 6, user is redirected to the ticket’s screen.

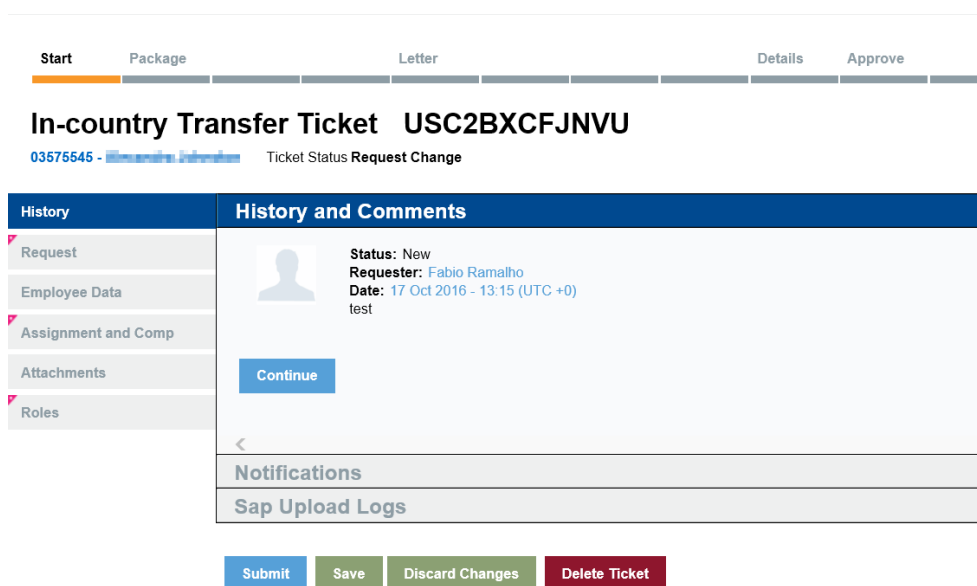


Figure 36 – Ticket ready

5.2.2.4 Execute events

There are two functions with the purpose of executing events:

- `Y_PCX_PROCESS_EVENT` – Field-based events (named “user events” within HCM P&F)
 - These can be events configured in HCM P&F or even custom events, as will be seen later;
- `Y_PCX_PROCESS_STANDARD_EVENT` – High-level “out of the box” events
 - This includes a full form check, submitting a form for approval (or send it back), cancelling a ticket, etc.

Function `Y_PCX_PROCESS_EVENT` signature is displayed in Table 6.

Parameter	Type	Data type	Description
IV_PCX_ID	Import	YPCX_PCX_ID	Ticket Identifier
IT_EVENTS	Import	YPCX_FIELD_EVENTS_T	Table with event(s) to execute
IT_DATA	Import	QISRTREQUEST_DATA	Table with form data
ET_DATA	Export	QISRTREQUEST_DATA	Updated form data
ET_ADDITIONAL_DATA	Export	QISRTREQUEST_DATA	Possible values calculated by the event(s)
ET_SUGGESTED_DATA	Export	QISRTREQUEST_DATA	Suggested values calculated by the event(s)
ET_MESSAGES	Export	YPCX_MESSAGE_TAB	List of messages
EV_IS_OK	Export	BOOLE_D	Indicator of success

Table 6 – Process Event – Signature

This function module relies on class method *PROCESS_EVENT* to execute the event and return its results. This method will evaluate the type of event (HCM P&F / Custom) and proceed with the necessary steps to execute them one by one. Figure 37 shows the execution flow of this function:

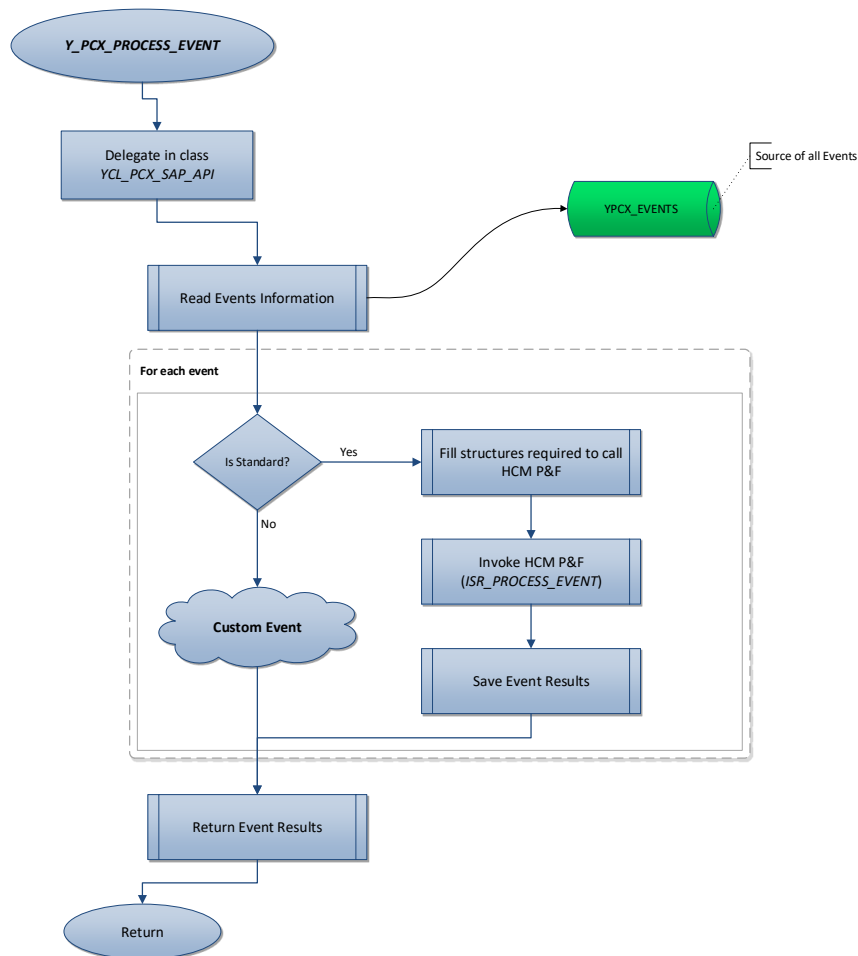


Figure 37 – Process Event – Flow of Execution

Function *Y_PCX_PROCESS_STANDARD_EVENT* has the following signature:

Parameter	Type	Data type	Description
IV_PCX_ID	Import	YPCX_PCX_ID	Ticket Identifier
IV_SAP_ADMIN	Import	PERSONID_EXT	Employee ID of the SAP Administrator
IV_CHECK	Import	BOOLE_D	Full Form Check indicator
IV_SAVE_DRAFT	Import	BOOLE_D	Save Draft indicator
IV_SEND	Import	BOOLE_D	Form Submission indicator
IV_CLOSE	Import	BOOLE_D	Ticket Closure indicator
IV_ASSIGN_POSTING	Import	BOOLE_D	SAP Admin assignment indicator
IV_DEASSIGN_POSTING	Import	BOOLE_D	SAP Admin de-assignment indicator
IV_CANCEL	Import	BOOLE_D	Cancel indicator
IV_DELETE	Import	BOOLE_D	Deletion indicator
IT_DATA	Import	QISRTREREQUEST_DATA	Form data
ET_MESSAGES	Export	YPCX_MESSAGE_TAB	List of messages generated
EV_IS_OK	Export	BOOLE_D	Indicator of success/failure

Table 7 – Process Standard Event – Signature

As displayed in Table 7 above, this function accepts several flags that indicate what action should be executed (each one corresponds to a type of event). This function module invokes method *PROCESS_STANDARD_EVENT*, providing these same flags. According to those values, the method will then communicate to HCM P&F the event that should be executed.

Figure 38 shows the execution flow of this function:

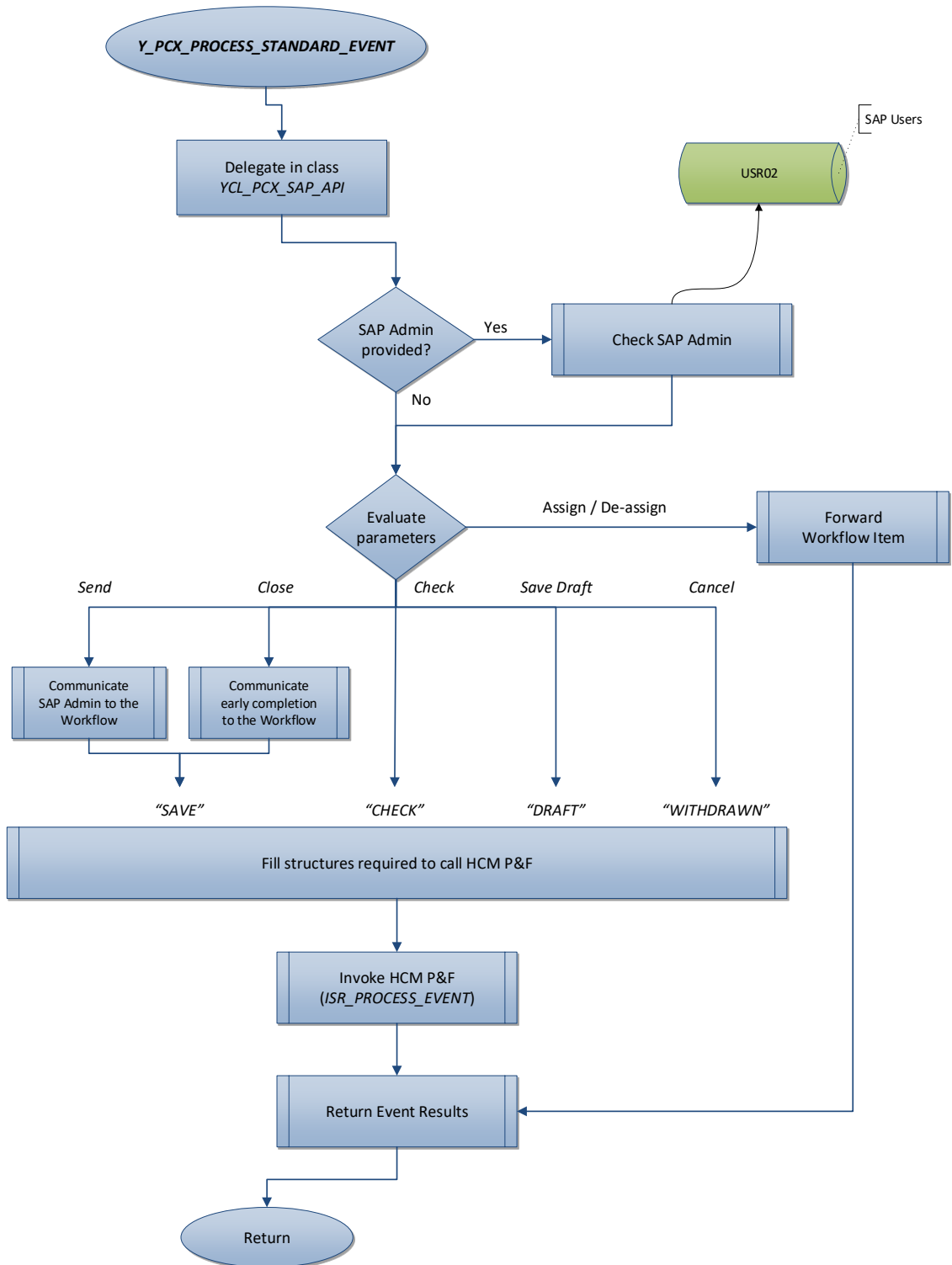


Figure 38 – Process Standard Event – Flow of Execution

5.3 SAP HCM Processes & Forms

HCM Processes & Forms was used to set the process/form in SAP. It is the component that controls all form-based configurations (for example: form fields, form events, etc.).

This section will approach all the configuration/developments done under the HCM P&F Framework.

5.3.1 P&F Solution Overview

A single process – *YP_PCX* – was created under the HCM P&F framework. This process is composed of a form scenario *YP01_PCX* and it's connected to workflow *YWF_PCX*, which controls the flow of the process. The following sections will focus on HCM P&F configuration, namely the process, the form scenario and their connection to SAP HR Master Data through Back-end Services. The workflow will be addressed later under section 5.4 SAP Workflow.

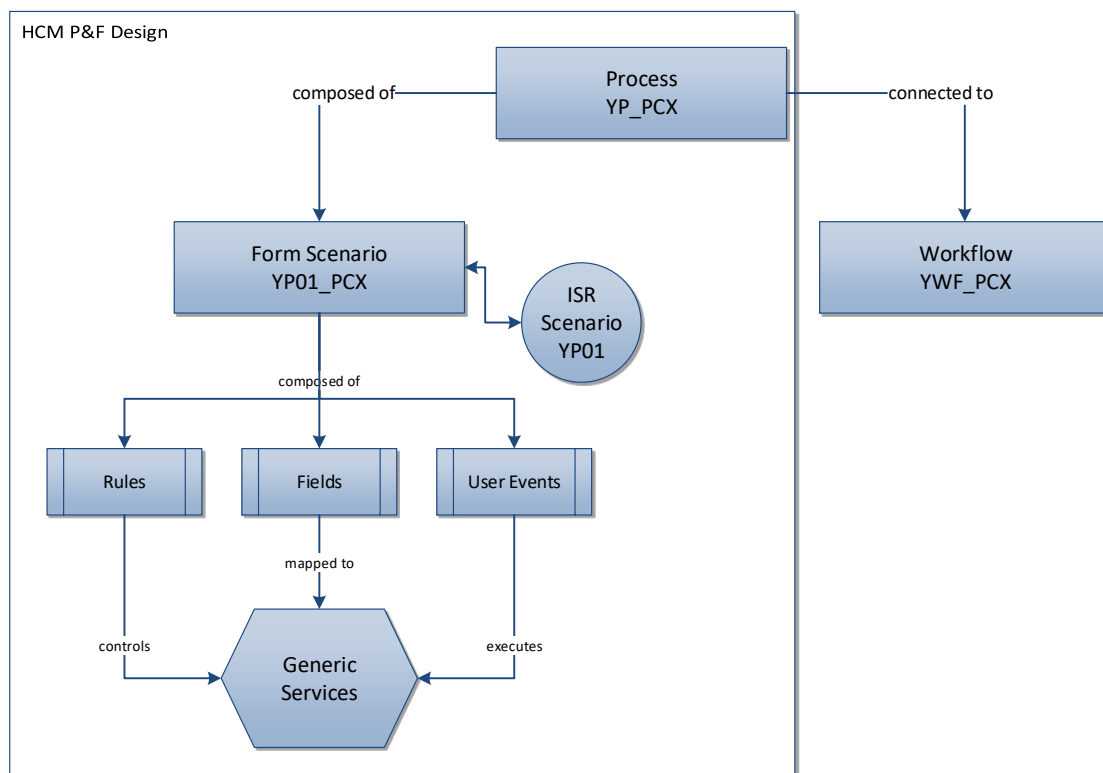


Figure 39 – HCM P&F Implementation Overview

This process was configured to have a start object – this assures the assignment of an employee to the process – and to tolerate error messages when submitting a form (the allowance of form submission is then handled by the web application):

Process	YP_PCX	ePCX Process
Validity Period		
Valid From	01.01.1800	Valid To 31.12.9999
Settings for Start Step		
<input type="checkbox"/> Show 'Forward to Expert' Button		
Send Variant	Tolerate Error Messages	
Process Type	Process with Start Object	

Figure 40 – HCM P&F Process – YP_PCX

5.3.2 P&F Form Scenario

Form Scenario *YP01_PCX* contains most part of the configuration related to HCM P&F:

- Form fields and their connection to Infotype fields;
- Rules to control which operations/validations should be carried;
- Events triggered when a user changes the value of a field.

5.3.2.1 Fields

The fields that were setup in the form scenario can be divided in three groups:

- **Infotype fields** – these are form fields directly connected to SAP infotypes. So, the users see what's the current data in those infotypes and when they change it, that will trigger the maintenance of those infotypes (once the process reaches SAP ERP). These are 90% of the fields on the form;
- **Search fields** – these fields are used as search filters to look possible values for the fields above;
- **Web fields** – these fields are only relevant for the web application, but need to be part of the process as well.

There are over 200 fields, so, in Table 8, we focus on the big picture by showing the SAP infotypes that are relevant (the 9### infotypes are customer-specific). The third column identifies infotypes that allow multiple entries on the form (e.g. a user may choose to enter several addresses).

Infotype	Description	Multi-Record
0000	Action (determined by the ticket type)	No
0001	Organizational Assignment	No
0002	Personal Data	No
0006	Addresses	Yes
0007	Work Schedule	No
0008	Basic Pay	No
0014	Additional Payments	No
0016	Contract Elements	No

Infotype	Description	Multi-Record
0021	Family Information	Yes
0022	Education	Yes
0041	Date Specifications	Yes
0077	Additional Personal Data (for US)	No
0094	Residence Status (for US)	No
9003	Documents	Yes
9004	Social Security	Yes
9006	Miscellaneous (Tax ID, etc.)	No
9008	Origin/Flying Point (for travels)	No
9009	Exit Interview	No
9017	Supplementary Contractor Data	No
9401	Local Employee Group (for France)	No
9408	Local Basic Pay (for France)	No

Table 8 – HCM P&F Form – Relevant Infotypes

Note that, depending on ticket type, fields may or may not be relevant. For instance, it wouldn't make sense to ask a user to enter an exit interviewer when the employee is being hired. For this reason, field security is handled by the web application.

5.3.2.2 User Events

User Events are used to trigger specific HCM P&F and/or SAP business logic for a restricted list of fields. For example, when a value of a field changes:

- That field or others may need to be validated;
- The possible values of dependent fields may need to be recalculated;
- The value of other fields may need to change.

So, there are two types of user events: “initialize” to (re)initialize field values and “check” for validations or fetching possible values. User Events are just responsible for triggering the execution of business logic. The logic itself is encapsulated in Back-end Services / Generic Services that will be detailed later. Several user events were selected from a preliminary list and created under HCM P&F, as long as the following rules were fulfilled:

1. Standard behaviour is the one expected by the web application;
2. Performance of going through HCM P&F framework is acceptable (vs. the duplication of business rules in custom layers).

Events that didn't respect these rules will be addressed under section 5.5 P&F External Events.

Table 9 contains the user events defined under HCM P&F. Each user event is associated with one field group – group of fields that are relevant for the user event. All events starting with I#### are related to infotype fields.

User Event	Description	Type	Ignore errors
I0001_CHECK_WERKS	Check Personnel Area	Check	No
I0002_CHECK_SSN	Check Social Security No.	Check	No

User Event	Description	Type	Ignore errors
I0006_CHECK	Check Address	Check	No
I0021_CHECK	Check Family Information	Check	No
I0022_CHECK	Check Education	Check	No
I0041_CHECK	Check Dates	Check	No
I9003_CHECK	Check Documents	Check	No
I9004_CHECK	Check Social Security	Check	No
FILL_REQUIRED_FIELDS	Default hidden SAP-only fields	Check	Yes
FILL_ALL_ADDITIONAL_DATA	Possible values for fields w/ no dependencies	Check	Yes
I0001_MOLGA_CHANGED	Possible values for fields dependent on country	Check	Yes
EFFECTIVE_DATE_CHANGED	Possible values for fields dependent on Effect. Date	Check	Yes
I0022_SLART_CHANGED	Possible values for fields dependent on Education Establishment	Check	Yes
I0000_FILL_MASSG	Possible values for Reason for Action	Check	No
I0001_FILL_BTRTL	Possible values for Personnel Subarea	Check	No
I0001_FILL_PERSK	Possible values for EE Group	Check	No
I0002_FILL_GBDEP	Possible values for State of Birth	Check	Yes
I0002_FILL_FPRCD	Possible values for IT2 Provincial Tax Code	Check	Yes
I0006_FILL_FPRCD	Possible values for IT6 Provincial Tax Code	Check	Yes
I0006_FILL_STATE	Possible values for Address State	Check	Yes
I0008_FILL_TRFGR	Possible values for Payscale Group	Check	Yes
I0008_FILL_TRFST	Possible values for Payscale Level	Check	Yes
I9401_FILL_PERSK	Possible values for Local EE Subgroup	Check	Yes
I0041_INITIALIZE	Refresh Dates	Initialize	Yes

Table 9 – HCM P&F User Events

Note that all infotypes which allow multiple records have a “check” event. This is required, so that each time a record is saved in the form, it is immediately checked. Additionally, there are several events to fetch possible values, marked to “ignore errors”, because the requirement is not to check their value, but just update the dependent fields.

5.3.2.3 Rules

HCM P&F Rules were created to control if specific validations should be carried, depending on the form and its status. Along with user events, this is a powerful tool to control what logic should be executed for each user action. These rules contain several conditions and their result is always a yes/no. Technically, these rules determine if:

- Infotype logic should be executed;
- A specific Back-end Service should be executed.

Taking this into consideration, these are the rules that were configured in HCM P&F:

- **1 rule for each infotype** that checks if any field was filled;
This will guarantee that validations are only executed if there is any data supplied. This is also crucial so that fields that are not being shown on the web application, are simply discarded.
- **1 rule for each Generic Service** to determine if its custom logic should be executed;
The condition will be detailed later when each Generic Service is presented.

This document will not get into details about the technical specificities of each rule. However, as an example, let's look at the rule *UPDATE_I0007*, which checks if logic related to Infotype 0007 – Work Schedule – should run:

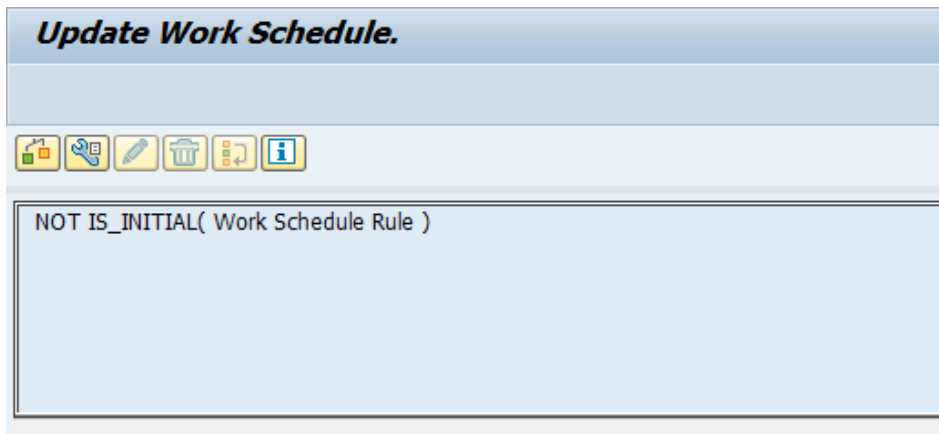


Figure 41 – HCM P&F Rule – Work Schedule

This is a very simple rule that checks if the form field “Work Schedule Rule” is filled or not.

5.3.3 P&F Back-end Services

Back-end Services contain business logic to be used by the process/form scenario. Technically, each Back-end Service corresponds to an ABAP class that encapsulates that same logic. A form

field is linked to a Back-end Service field and that's what makes it possible to apply default values or validations. Under Back-end Services, there are two categories:

- **Standard Services**, like *SAP_PA* (SAP Personnel Administration), which executes all the business logic related to an infotype. To accomplish this, standard services take advantage of the Decoupled Infotype Framework¹³, without duplicating any logic. The main benefit of these services is that they can be set up through configuration.
- **Generic Services**, used to add custom logic to the process.

Figure 42 gives an overview of how this occurs within HCM P&F:

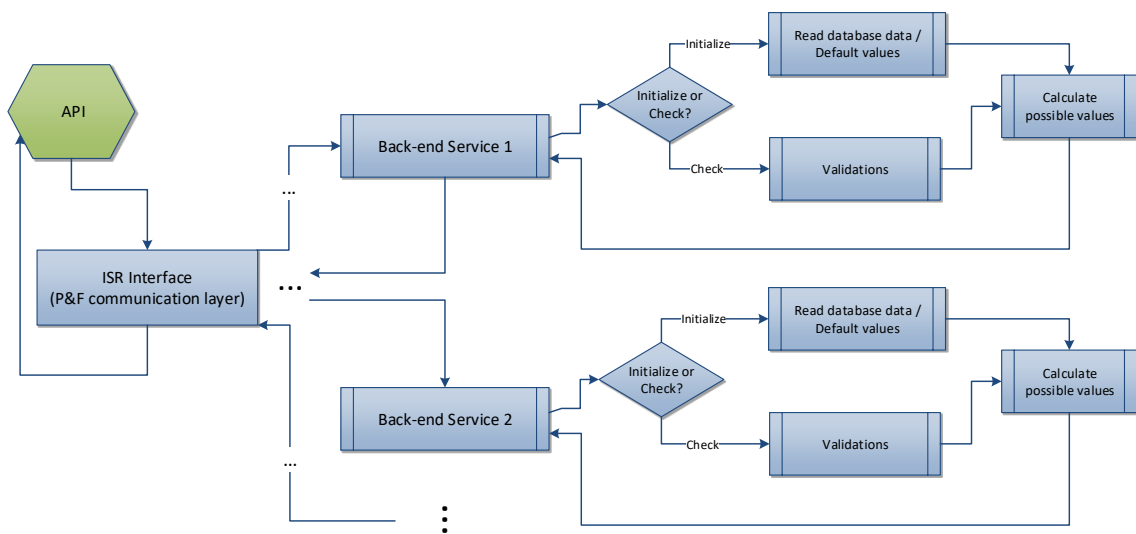


Figure 42 – HCM P&F – Back-end Services

5.3.3.1 SAP Personnel Administration (*SAP_PA*)

SAP_PA is the standard SAP Back-end Service for Personnel Administration Infotypes. It provides the means to default field values, fetch possible values and execute validations through configuration.

The configuration required to assign a form field to *SAP_PA* is as follows:

- Mapping the form field to an infotype field;
- Default (or not) the form field according to infotype field;
- Fetch the same possible values as the infotype screen does.

¹³ DITF is the new framework for infotype maintenance, which does away with the previous close link between business logic and the user interface. The infotype-specific business logic for decoupled infotypes is programmed in ABAP Objects classes (SAP SE, 2016v).

As an example, Figure 43 shows the configuration applied to infotype 0002 – Personal Data – fields:

Field Name	Name	Default Value	Input Help	Operation	Infotype	Subtype	O..	ScreenStruct	Field Name
I0002_ANRED	Title	3 SAP_PA: Value for Effective Date	3 SAP_PA: Value Help	Create	0002			HCMT_BSP_PA_XX_R0002	ANRED
I0002_ANZKD	Number of Children	3 SAP_PA: Value for Effective Date		Create	0002			HCMT_BSP_PA_XX_R0002	ANZKD
I0002_FAMDT	Date of Marriage	3 SAP_PA: Value for Effective Date		Create	0002			HCMT_BSP_PA_XX_R0002	FAMDT
I0002_FAMST	Marital Status	3 SAP_PA: Value for Effective Date	3 SAP_PA: Value Help	Create	0002			HCMT_BSP_PA_XX_R0002	FAMST
I0002_GBDAT	Date of Birth	3 SAP_PA: Value for Effective Date		Create	0002			HCMT_BSP_PA_XX_R0002	GBDAT
I0002_GBDEP	State of Birth	3 SAP_PA: Value for Effective Date	3 SAP_PA: Value Help	Create	0002			HCMT_BSP_PA_XX_R0002	GBDEP
I0002_GBLND	Country of Birth	3 SAP_PA: Value for Effective Date	3 SAP_PA: Value Help	Create	0002			HCMT_BSP_PA_XX_R0002	GBLND
I0002_GBORT	Birthplace	3 SAP_PA: Value for Effective Date		Create	0002			HCMT_BSP_PA_XX_R0002	GBORT
I0002_GESCH	Gender	3 SAP_PA: Value for Effective Date	3 SAP_PA: Value Help	Create	0002			HCMT_BSP_PA_XX_R0002	YGESCH
I0002_MIDNM	Middle Name	3 SAP_PA: Value for Effective Date		Create	0002			HCMT_BSP_PA_XX_R0002	YYMIDNM
I0002_NACHN	Last Name	3 SAP_PA: Value for Effective Date		Create	0002			HCMT_BSP_PA_XX_R0002	NACHN
I0002_NAME2	Name at Birth	3 SAP_PA: Value for Effective Date		Create	0002			HCMT_BSP_PA_XX_R0002	NAME2
I0002_NATI2	Second Nationality	3 SAP_PA: Value for Effective Date	3 SAP_PA: Value Help	Create	0002			HCMT_BSP_PA_XX_R0002	NATI2
I0002_NATIO	Nationality	3 SAP_PA: Value for Effective Date	3 SAP_PA: Value Help	Create	0002			HCMT_BSP_PA_XX_R0002	NATIO
I0002_PERID	Social Security Number	3 SAP_PA: Value for Effective Date		Create	0002			HCMT_BSP_PA_XX_R0002	YPPERID
I0002_SPRSL	Common Language	3 SAP_PA: Value for Effective Date	3 SAP_PA: Value Help	Create	0002			HCMT_BSP_PA_XX_R0002	SPRSL
I0002_VORNA	First Name	3 SAP_PA: Value for Effective Date		Create	0002			HCMT_BSP_PA_XX_R0002	VORNA

Figure 43 – SAP_PA – Infotype 0002 configuration

Note that all fields are defaulted with current data in the infotype (column “Default value”) and fields that should have a list of possible values are flagged in column “Input Help”.

Several infotypes were connected to SAP_PA, each one with their specificities:

Infotype	Subtype	Operation
0000		Change
0001		Create
0002		Create
0007		Create
0008	0	Create
0014	2003	Create
0016		Create
0077		Create
0094		Create
9006		Create
9008		Create
9009	*	Create
9401		Create
9017		Create
9408		Create

The column “operation” indicates the action that will be done for each infotype, despite being just a simulation for validation purposes. So, a “Create” operation means a new record is created for that infotype so that its data is validated, but nothing is persisted.

5.3.3.2 Generic Services

Due to special requirements, some infotypes could not be handled through *SAP_PA*. In these cases, SAP SE provides Generic Services to implement custom logic that will be automatically called by the HCM P&F framework, similarly to standard services.

Generic Services are created under BAdI *HRASROOGEN_SERVICE_BASIC* (so-called **Basic Generic Services**) or *HRASROOGEN_SERVICE_ADVANCED* (so-called **Advanced Generic Services**) of Enhancement Spot *HRASROOGENERIC_SERVICES*. These BAdIs implement interfaces *IF_HRASROOGEN_SERVICE* and *IF_HRASROOGEN_SERVICE_ADVANCED*, respectively.

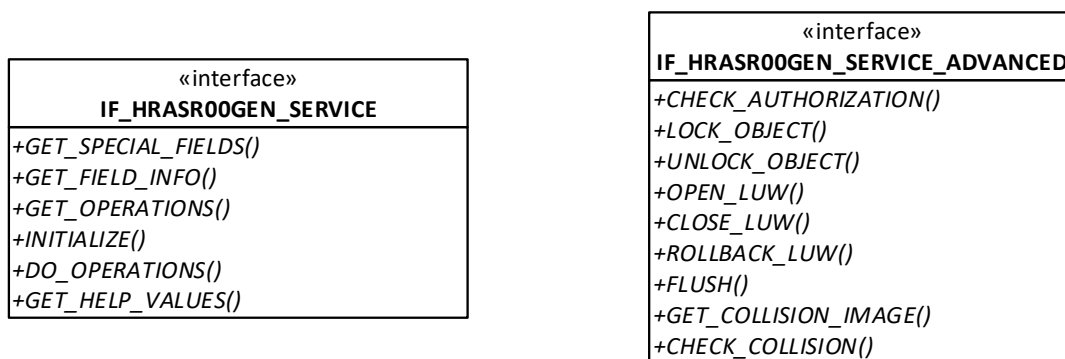


Figure 44 – Generic Services Interfaces

In this project, all Generic Services were created under BAdI *HRASROOGEN_SERVICE_BASIC*, since it contains the necessary methods to handle fields’ defaults, validations and possible values. *HRASROOGEN_SERVICE_ADVANCED* is not used, because there’s no automatic persistence of HCM P&F form data into the SAP HR database. That task is done by SAP Administrators with help of assisted posting features, as will be detailed later.

Considering this, three Basic Generic Services were created:

- Additional Functions
- Date Specifications
- Multiple Records

5.3.3.2.1 Additional Functions

This Generic Service is valid for several fields of different infotypes and it was created with the purpose of adding custom logic:

- Execute validations that are not supported by standard SAP service;
- Gather default values / possible values for fields that are not part of an infotype;
- Gather default values / possible values for custom fields.

Technically, it is linked to BAdI Implementation *YPCX_GS_ADDFUNCTIONS*. This BAdI Implementation uses class *YCL_PCX_GS_ADDFUNCTIONS*, where all the logic is encapsulated.

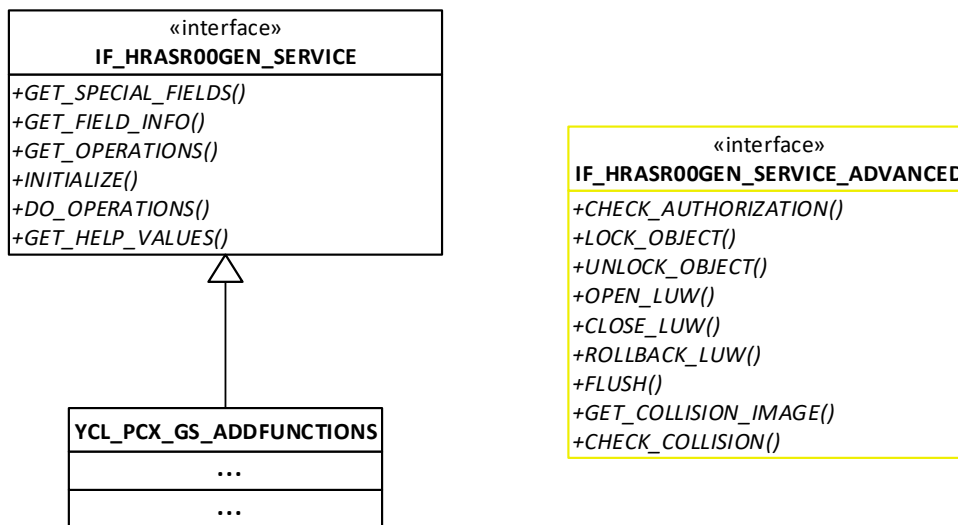


Figure 45 – Generic Service: Additional Functions

Special Fields

Every Generic Service contains a list of special fields that are not changed during its runtime. The main benefit of this functionality is that, before using a Generic Service, we already know what is affected and what is not. These fields are typically used for reading purposes in order to validate dependent fields. For example, the country of assignment is a special field. Its value is read to find out if a country’s specific logic should be executed.

The special fields required to execute Generic Service *YPCX_GS_ADDFUNCTIONS* are:

- *EFFECTIVE_DATE* (Effective Date)
- *PERSONID* (GIN)
- *USER_EVENT* (User Event Name)
- *PCX_ID* (Ticket ID)
- *PCX_TYPE* (Ticket Type)
- *PCX_COUNTRY* (Country)

Default Values

SAP_PA Service only handles Personnel Administration infotype fields. Therefore, fields that are not part of an infotype need to be handled with specific logic. The fields defaulted by this Generic Service are displayed in Table 10:

Field Name	Default value
CENTRAL_PERSON	Employee’s Central Person Number
COMPANY	Employee’s Company (based on Org Unit)
COST_ELEMENT	Employee’s Cost Center
DIRECT_MANAGER	Employee’s HRPS Manager
F_AREA	Employee’s Area
F_FINANCE_SYSTEM	Employee’s Finance System

Field Name	Default value
F_GEOMARKET	Employee's Geomarket
F_JOB_DISCIPLINE	Employee's Job Discipline (through Job)
F_JOB_GROUP	Employee's Job Group (through Job)
F_SUBSEGMENT	Employee's Subsegment
I0024_PTE	Employee's PTE Rating
I0024_SETC	Employee's SETC
I2002_SUBTY	Employee's Attendance Type
LOA_MANAGER	Employee's LOA Manager (= HRPS Manager)
LOA_N2_MANAGER	Employee's LOA N+2 Manager (= HRPS Manager's Manager)
MANAGED_ORGUNITS	Organizational Units managed by employee
MANAGER_POSITION	True or false, considering if employee is a manager
MASSN	SAP Action based on the Ticket Type
OLD_SAL_MAX	Employee's current salary maximum (according to Salary Structure)
OLD_SAL_MAX_CUR	Employee's current salary maximum currency (according to Salary Structure)
OLD_SAL_MIN	Employee's current salary minimum (according to Salary Structure)
OLD_SAL_MIN_CUR	Employee's current salary minimum currency (according to Salary Structure)
PERNR	Employee's Active Personnel Number (based on GIN)
REVIEWING_MANAGER	Employee's N+2 Manager (= HRPS Manager's Manager)
WBS_ELEMENT	Employee's WBS Element (Attendance Type ABCA if exists, else Attendance Type ABDC)
WC_CITY	Employee's Work Location City
WC_COUNTRY	Employee's Work Location Country
WORK_CENTER	Employee's Work Location

Table 10 – Generic Service Additional Functions – Default values

All the employee data above is calculated based on the effective date, which corresponds to the system's current date during the creation of the ticket.

Possible values

In some cases, *SAP_PA* does not meet the requirements in terms of possible values. In regards to this Generic Service, these are the fields for which possible values are calculated:

Field Name	Possible values
COMPANY	Company/Business Segment possible values
CSP_MP_CUR	Currency possible values
F_FINANCE_SYSTEM	Finance System / Origin System possible values (dom. YORG_SYS)
I0001_BTRTL	Personnel Subarea possible values

Field Name	Possible values
I0001_PERSG	Employee Group possible values (in addition, these values are filtered by Country Grouping, as opposed to the standard behaviour)
I0001_PERSK	Employee Subgroup possible values
I0008_ANCUR	Currency possible values
I0008_PREAS	Pay Reason possible values (except reason 10)
I0008_WAERS	Currency possible values
I0024_PTE	PTE Rating possible values
I0024_SETC	SETC Employee possible values
I9017_CNT_RATE_CUR	Currency possible values
MASSG	Reason for Action possible values (based on SAP Action – MASSN)
MASSN	SAP Action possible values
OLD_SAL_MAX_CUR	Currency possible values
OLD_SAL_MIN_CUR	Currency possible values
PROP_SAL_MAX_CUR	Currency possible values
PROP_SAL_MIN_CUR	Currency possible values
WC_REGION	Work Location Region possible values, based on Ticket's Country

Table 11 – Generic Service Additional Functions – Possible values

Operations

Operations of a Generic Service are executed once a check is performed (remember that a check can be a full form check or just a user event to validate a list of fields and fetch their possible values). The operations provided by this Generic Service are identified in Table 12:

Operation	Description
CHECK_I0001	Execute additional validations for Infotype 0001
CHECK_I0002	Execute additional validations for Infotype 0002
CHECK_I0014	Execute additional validations for Infotype 0014
CHECK_I0016	Execute additional validations for Infotype 0016
CHECK_I0021	Execute additional validations for Infotype 0021
CHECK_MANAGERS	Validate if Direct Manager is not one of the Functional Managers
CHECK_NO_SPEC_CHARS	Validate if a field contains invalid characters
CHECK_SSN	Execute additional validations for Social Security Number
CONVERT_AMOUNTS	Convert salary amounts
FILL_I2002_SUBTY	Determine the subtype to be suggested for Infotype 2002
FILL_I9009_SUBTY	Determine the subtype to be suggested for Infotype 9009
FILL_ITBLD	Determine country indicator for an address
FILL_JUPER	Determine Legal Person based on Company Code

Operation	Description
FILL_KOKRS	Determine Controlling Area based on Company Code
FILL_WAERS	Use the annual salary's currency for other fields
FILL_YYFUN	Determine SAP Finance System based on GAR Origin System
HANDLE_PERNR	Validate Personnel Number, regenerate it if already hired

Table 12 – Generic Service Additional Functions – Operations

5.3.3.2.2 Date Specifications

One of the requirements of this product was that specific dates would be shown on the form. In other words, not all dates that are stored in SAP infotype 0041 – Date Specifications – are relevant for this process. This is the main reason why infotype 0041 cannot be handled through standard *SAP_PA* service.

Considering this, a Generic Service was created to handle this infotype alone. These are the main tasks performed by *YPCX_GS_DATE_SPECIFICATIONS*:

- Read only specific dates from Infotype 0041 (e.g. hire date, position seniority, etc.).
- Default dates when relevant fields in the form change:
 - For instance, if the employee's annual salary changes, salary seniority date is automatically updated to the effective date.
- Execute Infotype 0041's validations by taking advantage of same logic used by *SAP_PA* service (i.e., DITF business logic).

Similarly to the “Additional Functions” Generic Service, this is also linked to a BAdI Implementation called *YPCX_FS_DATE_SPECIFICATIONS*, which relies on class *YCL_PCX_GS_DATE_SPECIFICATIONS*:

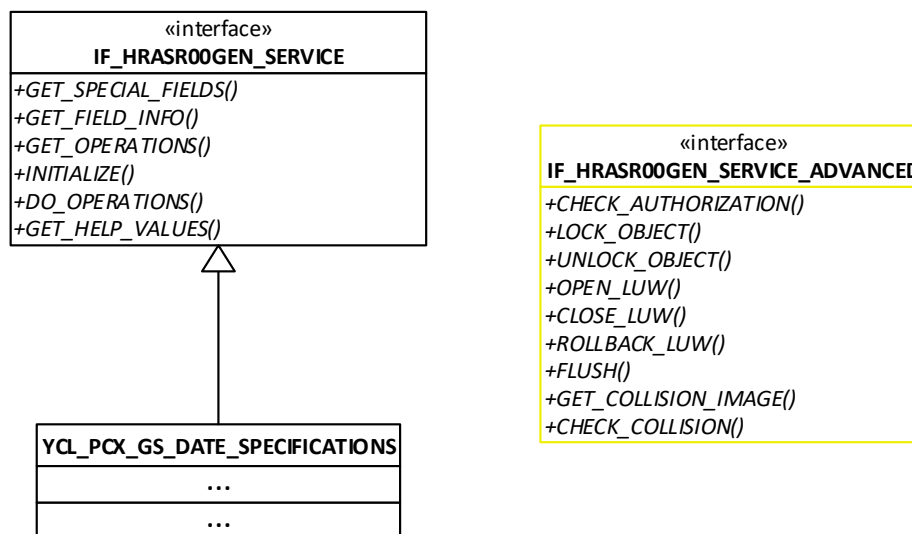


Figure 46 – Generic Service: Date Specifications

Special fields

The special fields required to execute Generic Service *YPCX_GS_DATE_SPECIFICATIONS* are:

- *EFFECTIVE_DATE* (Effective Date)
- *PCX_ID* (Ticket ID)
- *PCX_COUNTRY* (Country)
- *PERNR* (Personnel Number)
- *MASSN* (SAP Action)
- *MASSG* (Reason for Action)
- *USER_EVENT* (User Event Name)

Default values

Infotype 0041 is structured statically, with one field for each date type, as displayed in Figure 47:

Component	RT...	Component type	Data Type	Length	Deci...	Short Description
.INCLUDE	<input type="checkbox"/>	PS0041	□□	0	0	HR Master Record: Infot...
DAR01	<input type="checkbox"/>	DATAR	CHAR	2	0	Date type
DAT01	<input type="checkbox"/>	DARDT	DATS	8	0	Date for date type
DAR02	<input type="checkbox"/>	DATAR	CHAR	2	0	Date type
DAT02	<input type="checkbox"/>	DARDT	DATS	8	0	Date for date type
DAR03	<input type="checkbox"/>	DATAR	CHAR	2	0	Date type
DAT03	<input type="checkbox"/>	DARDT	DATS	8	0	Date for date type
DAR04	<input type="checkbox"/>	DATAR	CHAR	2	0	Date type
DAT04	<input type="checkbox"/>	DARDT	DATS	8	0	Date for date type
DAR05	<input type="checkbox"/>	DATAR	CHAR	2	0	Date type
DAT05	<input type="checkbox"/>	DARDT	DATS	8	0	Date for date type
DAR06	<input type="checkbox"/>	DATAR	CHAR	2	0	Date type
DAT06	<input type="checkbox"/>	DARDT	DATS	8	0	Date for date type

Figure 47 – Infotype 0041 – Table structure

However, the form being built here should follow the multiple records pattern, which means there should only be two fields: date type and date, where users may add multiple lines.



Key Dates			
Action	Date Type	Date	
 	Initial Hire Date	11 May 1992	
 	Benefits Seniority	11 May 1992	
 	Area Seniority	11 May 1992	

Figure 48 – Dates on the form

Considering this, there are just two fields defaulted by this Generic Service:

- Date Type – Form field *10041_DATAR*;
- Date – Form field *10041_DARDT*.

These fields will have an index per date type. Additionally, for every index, two more fields will exist in order to provide the option of deleting a date:

- Delete flag – Form field *10041_DELETE*;
- Reason for deletion – Form field *10041_REASON_DEL*.

Several rules were developed to default/recalculate dates automatically, while the user is interacting with the form, according to specific business requirements:

1. On ticket creation, dates will be read from Infotype 0041 (i.e. current employee data). For “New Hire” or “Contractor Hire”, there are no data available yet, so the relevant dates will be defaulted to current date.
2. If Employee Group changes on a Re-hire, Organizational Transfer or International Transfer, Employee Group Seniority Date will be updated with the selected effective date.
3. If Grade changes, Grade Seniority Date will be updated with the selected effective date.
4. If Salary changes, Salary Seniority Date will be updated with the selected effective date.
5. If Organizational Unit or Job changes, Position Seniority Date will be updated with the selected effective date.
6. In Termination tickets, Last Day of Employment and First Day of Non-Employment are defaulted with the selected effective date and the next day, respectively.

Possible values

No possible values are configured in this Generic Service. The allowed date types are manually configured in the form scenario.

Operations

The only operation configured for this Generic Service is *UPDATE_I0041* with the purpose of validating the record to be inserted in infotype 0041. Data is not persisted.

5.3.3.2.3 Multiple Records

As described under 5.3.2 P&F Form Scenario, the form contains several fields that correspond to multi-record infotypes. In other words, users can add several records to the same infotype. There are 5 infotypes in this situation:

- Infotype 0006 – Addresses
- Infotype 0021 – Family Information
- Infotype 0022 – Education
- Infotype 9003 – Documents
- Infotype 9004 – Local Social Security

Also, for each one of these infotypes, the user must be able to:

- Insert a new record
- Update an existing record
- Delete (delimit) an existing record

This functionality cannot be achieved through standard service *SAP_PA* due to limitations in the framework. This is documented in SAP Note 1043692: “You may have multiple infotype table controls on your form but then you cannot perform ‘*DELETE*’ operation on infotype records by using the check box approach. It is recommended to use only one infotype table control on the form when you intend to perform *DELETE* operation on infotype records” (SAP SE, 2010a).

For this reason, *YPCX_GS_MULTIPLE_RECORDS* was created. It is responsible for handling operations of the infotypes mentioned above. It has however no responsibility regarding default/possible values, since that can still be provided by *SAP_PA*. Due to the similarities between these infotypes and the fact that it’s likely to have new multi-record infotypes added in the future, this Generic Service was built with focus on dynamism. This is, it will accept an unlimited number of fields and the operations executed will rely on infotype’s logic.

YPCX_GS_MULTIPLE_RECORDS should have a 1:1 mapping between itself and an infotype. For example, an instance of this Generic Service should be configured for infotype 0006, another one for infotype 0021 and so on.

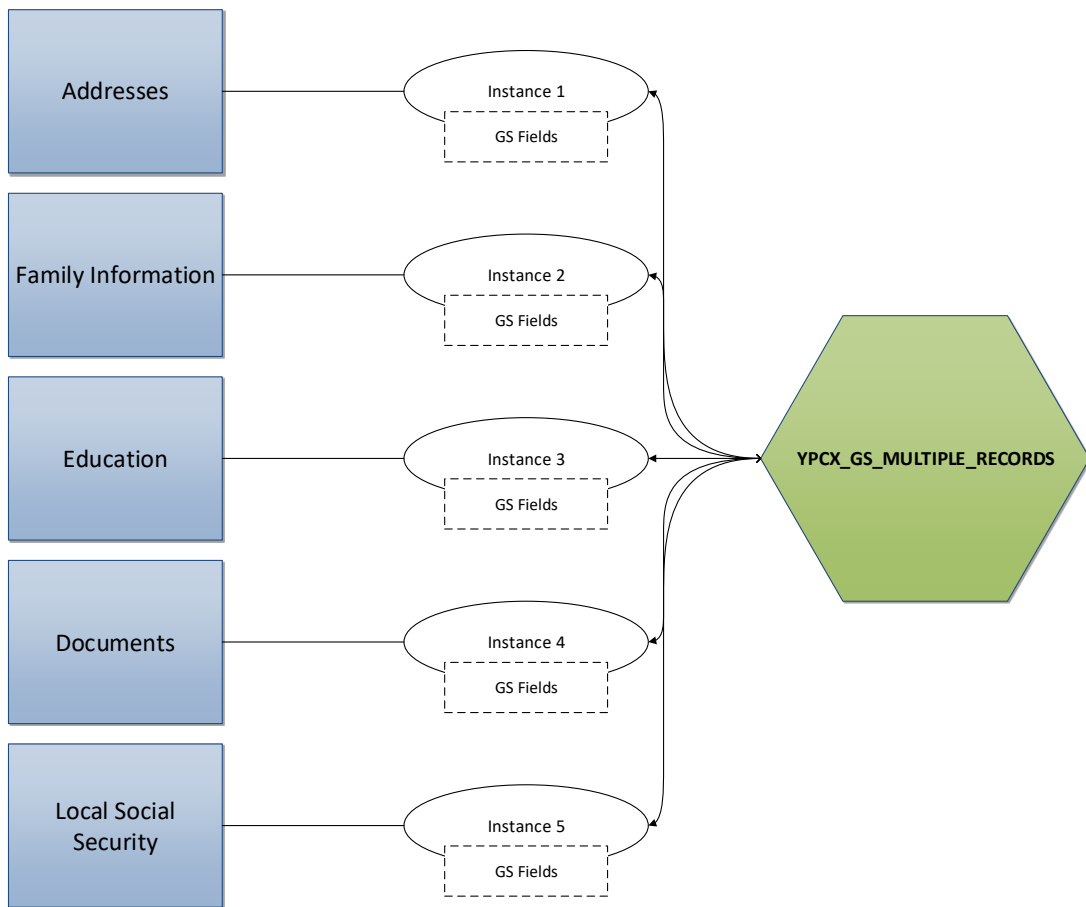


Figure 49 – Mapping between Infotypes and Multiple Records GS

As other Generic Services, this is linked to a BAdI Implementation – *YPCX_GS_MULTIPLE_RECORDS*, which relies on class *YCL_PCX_GS_MULTIPLE_RECORDS*.

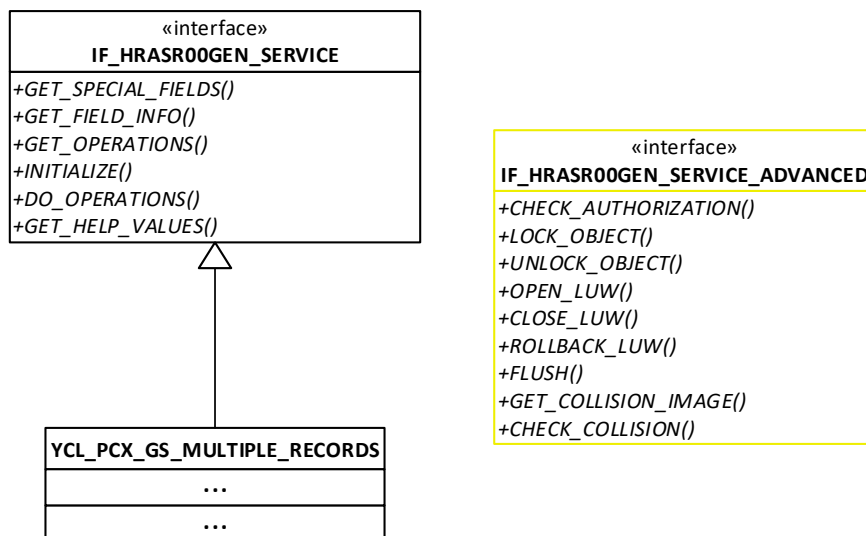


Figure 50 – Generic Service: Multiple Records

Special fields

The special fields required to execute Generic Service *YPCX_GS_MULTIPLE_RECORDS* are:

- *EFFECTIVE_DATE* (Effective Date);
- *PERNR* (Personnel Number).

Default values

There are no values defaulted by this Generic Service. However, for every index, two Generic Service fields will exist to provide the option of deleting a record:

- Delete flag – *DELETE_DATASET* (dynamically assigned to a field through configuration, depending on the infotype);
- Reason for deletion – *REASON_DELETE* (dynamically assigned to a field through configuration, depending on the infotype).

Possible Values

No possible values are configured in this Generic Service.

Operations

The only operation configured in this Generic Service is called *UPDATE_TC3_INFITY*. Its purpose is to validate all the records added in each infotype. To accomplish this, the Generic Service delegates this on the Decoupled Infotype Framework, where each infotype will be validated according to their rules. Data is not persisted.

5.3.4 P&F Framework Enhancements

In order to adjust HCM Processes & Forms to the specific requirements of this project, some enhancements were made to the framework. Those enhancements are described in the following two sections.

5.3.4.1 Form Check

When a user finishes filling the form and submits it for approval, a full check is carried to validate all fields according to SAP rules. However, during this process, the HCM P&F framework also recalculates the possible values for all fields in the form (this was already pointed in Figure 42 – HCM P&F – Back-end Services). Considering the application requirements, this is completely unnecessary and above all, it is a lengthy process that even surpasses the time used for validations. Possible values should only be re-calculated through events triggered by user action on a particular field. Therefore, in order to decrease the full check execution time, an enhancement was made to the framework's standard code.

The SAP ABAP Enhancement Framework provides 3 types of class-component enhancements: pre-exit, post-exit and overwrite-exit, which respectively allows developers to add code before the method execution, after or simply replace it (SAP SE, 2016t).

In class *CL_HRASROO_DISPATCHER* – HCM P&F Dispatcher – an adjustment was made to the method *IF_HRASROO_DISPATCHER~CHECK*, the wrapper to form checks. SAP SE already provides a parameter that suits our purpose – *SKIP_HELP_VALUES_COMPUTATION* – but it was never activated in previous implementations. So, an overwrite-exit was created in order to activate the parameter *SKIP_HELP_VALUES_COMPUTATION* according to what is set by the API. The remaining logic of the original method was then replicated in the new one. Note that this enhancement does not disrupt other HCM P&F implementations, since it's simply a switch turned on/off by the API only used in the context of this process.

Figure 51 shows the execution flow after these changes were applied.

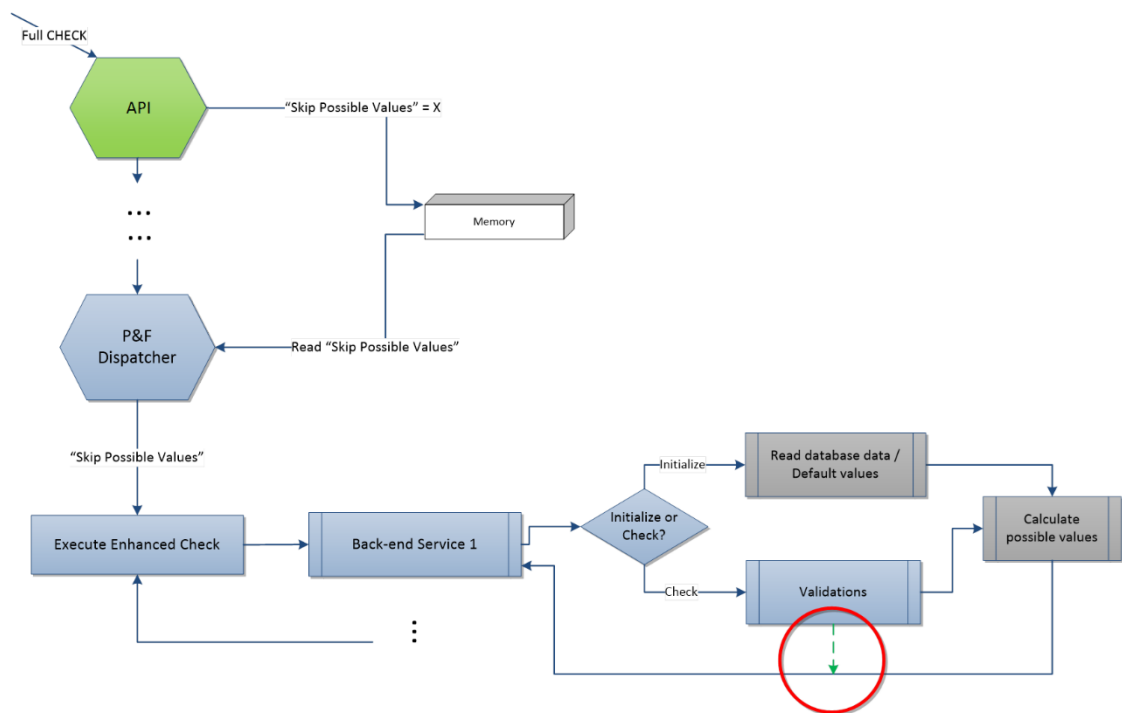


Figure 51 – HCM P&F – Enhanced Form Check

5.3.4.2 HCM P&F Messages

One of the product's requirements is that error/warning messages are shown next to the field that caused them. To accomplish this, the API must expose messages with a fieldname attached to them.

Technically, messages need to be assigned to a fieldname and an index, because in case of a multiple record, that message should be associated with the correct record/line. The ISR framework already offers a structure that provides the means to do this. ISR messages are returned under the structure type *BAPIRET2*:

Component	Component Type	Description
TYPE	BAPI_MTYPE	Message type: S Success, E Error, W Warning, I Info, A Abort
ID	SYMSGID	Message Class
NUMBER	SYMSGNO	Message Number
MESSAGE	BAPI_MSG	Message Text
LOG_NO	BALOGNR	Application log: log number
LOG_MSG_NO	BALMNR	Application log: Internal message serial number
MESSAGE_V1	SYMSGV	Message Variable 1
MESSAGE_V2	SYMSGV	Message Variable 2
MESSAGE_V3	SYMSGV	Message Variable 3
MESSAGE_V4	SYMSGV	Message Variable 4
PARAMETER	BAPI_PARAM	Parameter Name
ROW	BAPI_LINE	Lines in parameter
FIELD	BAPI_FLD	Field in parameter
SYSTEM	BAPILOGSYS	Logical system from which message originates

Table 13 – ISR Structure – *BAPIRET2*

Note the components *ROW* and *FIELD* that, respectively, give the line and the name of the field to which the message relates.

HCM P&F is part of the ISR framework. However, there's no standard functionality to extract these two parameters from HCM P&F messages (structure *HRBAS_MESSAGE*) into ISR messages.

Component	Component Type	Description
CAUSE	HRBAS_MESSAGE_CAUSE	Reason for Message Output
DETAIL_LEVEL	BALLEVEL	Application Log: Level of detail
MSGTY	SYMSGTY	Message Type
MSGID	SYMSGID	Message Class
MSGNO	SYMSGNO	Message Number
MSGV1	SYMSGV	Message Variable 1
MSGV2	SYMSGV	Message Variable 2
MSGV3	SYMSGV	Message Variable 3
MSGV4	SYMSGV	Message Variable 4
FIELD_LIST	HRBAS_FIELD_TAB	Field List
CONTEXT	IF_HRBAS_MESSAGE_CONTEXT	Message Context

Table 14 – HCM P&F Structure – *HRBAS_MESSAGE*

Note the component *FIELD_LIST* that contains all fields associated with each message. Unfortunately, as previously stated, those fieldnames are not exported to ISR scope. Therefore, an enhancement was created to add this new feature. This enhancement is composed of two overwrite-exits in class *CL_IM_HRASROOISR* – the class at the level above the HCM P&F Dispatcher and responsible for the connection between ISR and HCM P&F:

1. Method *GEN_MSGS_FOR_MANDATORY_FIELDS*

GEN_MSGS_FOR_MANDATORY_FIELDS is called after the execution of an operation (e.g. validate the form). It's responsible for generating additional messages regarding mandatory fields and retrieving field information to replace message texts.

The method was enhanced in order to add a masked index to each message returned by HCM P&F. During runtime, right after this method, duplicate messages are removed by HCM P&F standard logic. Therefore, with this enhancement it's assured that the same message can be shown for different indexes/lines instead of being discarded.

2. Method *GET_MESSAGES_FOR_ISR*

GET_MESSAGES_FOR_ISR is called after the method mentioned above and it's responsible for mapping messages from the HCM P&F structure (*HRBAS_MESSAGE*) to the ISR exporting structure (*BAPIRET2*).

The method was enhanced in order to map the form field and the index for every message, meaning that the components *ROW* and *FIELD* are now filled accordingly to what was found in HCM P&F messages.

5.4 SAP Workflow

5.4.1 Technical Overview

As pointed out earlier, a single workflow – *YWF_PCX* – was created in SAP ERP, valid for all ticket types. This workflow complements HCM Processes & Forms with a flow and the responsible agents.

It consists of several interactive steps where communication between the workflow and the persistence layer of runtime data is constantly occurring, either by reading, checking or submitting form data, as well as background steps where data considered irrelevant to the user is handled.

Each step of the workflow is connected to what is called a **workflow task**. These workflow tasks work as wrappers to the step's logic. Usually, this logic is encapsulated in a utility class, with a 1:1 relationship between a workflow task and a method. Also, note that a workflow task can be reused in several steps.

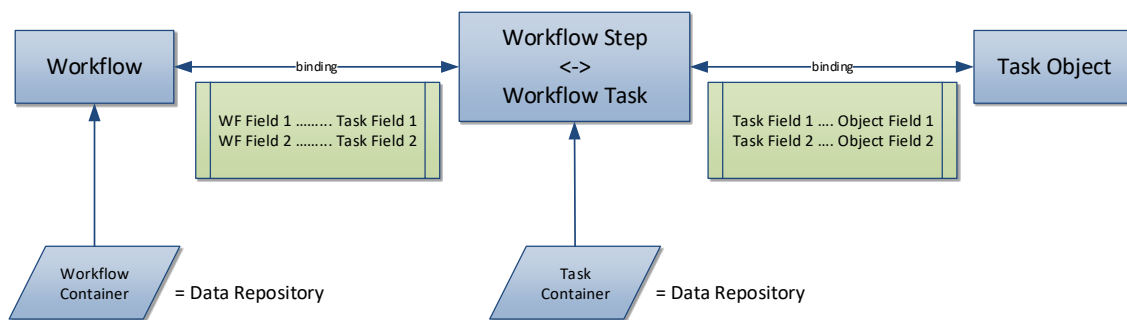


Figure 52 – SAP Workflow Tasks

In this particular case, class `YCL_PCX_WORKFLOW_UTILS` was created to hold all the custom logic required for each step. This will be detailed later along with each step.

As seen in section 4.3.3 SAP Workflow, this workflow has 3 major steps for user interaction:

- Edit & Submit Form Data
- SAP Assisted Posting
- Review Form Data (Manual Step)

However, these interactive steps are supported by several background tasks that run before/after them. Figure 53 gives a technical view of the workflow as it was built within SAP ERP.

It's possible to see the interactive tasks (in blue) and the background tasks (in green).

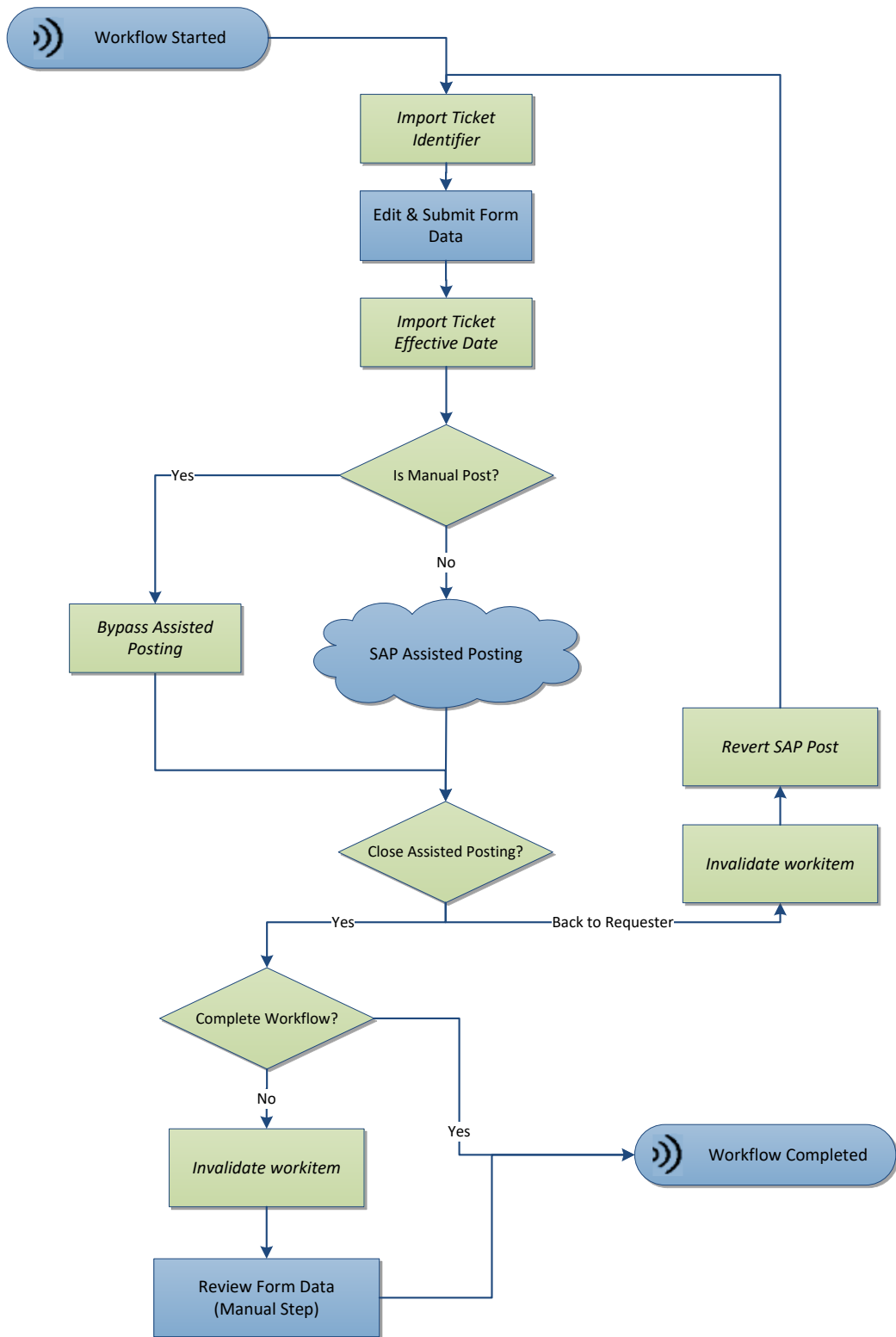


Figure 53 – SAP Workflow

Starting from the top, the workflow is launched when the user chooses to create a ticket.

Technically, this means that the API is invoked to start a process. The API will trigger an event which will create the persistence layer of runtime data – Process Object – and consequently, the workflow will start. To accomplish this, the workflow must react to an event launched by the HCM P&F framework (namely, standard class *CL_HRASR00_WF_PROCESS_OBJECT*). Figure 54 shows the configuration that was applied.

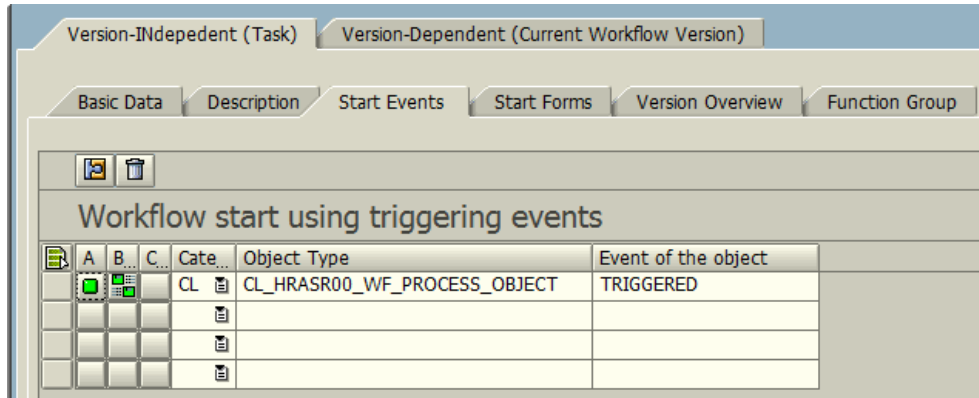


Figure 54 – SAP Workflow Start

The binding between event and the workflow is very simple, containing data that uniquely identifies the ticket:

Event Container	Workflow Container
_EVT_CREATOR	_WF_INITIATOR
_EVT_OBJECT	PROCESS_OBJECT
_EVT_OBJECT.LEADING_SCENARIO	FORM.FORM_SCENARIO
_EVT_OBJECT.LEADING_SCENARIO_VERSION	FORM.FORM_SCENARIO_VERSION

Table 15 – SAP Workflow Start Event – Binding

While the user is interacting with the form (checking data, submitting data, etc.), the workflow must be constantly communicating with the data layer. To ensure the communication/synchronization between the workflow and the process object, SAP SE offers the standard program exit *CL_HRASR00_POBJ_WF_EXIT*.

As Figure 55 will show, the workflow and the program exit have a two-way communication:

- The workflow can export data to update the process object;
- The workflow can read data from the process object.

In the same way, the program exit’s class *CL_HRASR00_POBJ_WF_EXIT* will communicate with the process object, stored as a Case Management document.

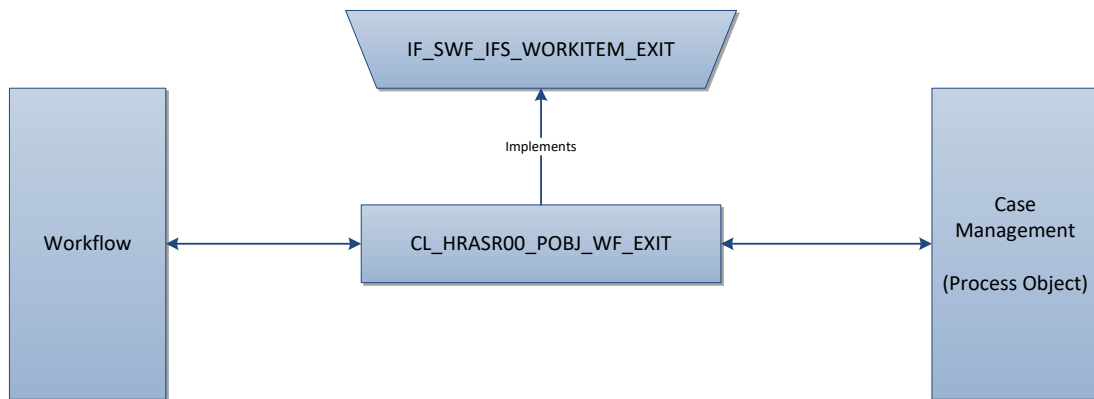


Figure 55 – HCM P&F Program Exit

In the following sections, each workflow step will be approached individually.

5.4.2 Workflow Steps

5.4.2.1 Import Ticket Identifier

This is a background step (no action from end user) and its purpose is to fetch form data that is relevant for the execution of subsequent steps. To accomplish this behaviour, a standard task is offered by SAP SE within the HCM P&F framework (*TS17900110 – Import Form Container -> WF Container*). Technically, this task imports data from the form container into the workflow container. In this particular case, it imports the ticket identifier.

The binding between the workflow and this step is configured as displayed in Table 16:

Field Name	Type	Value	Description
FORM	Export	YP01_PCX 0000	Form Scenario (identifies the process/form)
FORM_FIELD_NAME_1	Export	PCX_ID	Field name to be read
FORM_FIELD_VALUE_1	Import		Ticket ID

Table 16 – SAP Workflow Step – Import Ticket Identifier

5.4.2.2 Edit & Submit Form Data

This is an interactive step, meaning it will depend on user action in order to advance the workflow. On this step, users are able to edit the form, check data and save it. Technically, this step is again connected to a standard HCM P&F task (*TS17900100 – Process Form*). While on this step, the ISR interface is used to communicate with the process object through events triggered by the user.

The binding between the workflow and this step is configured as displayed in Table 17:

Field Name	Type	Value	Description
FORM	Export	YP01_PCX 0000	Form Scenario (identifies the process/form)
FORM_SCENARIO_STAGE	Export	YP01_PCX_A	Form Scenario Step
SAVE_DRAFT_BUTTON_VISIBLE	Export	X	Allows the possibility of a draft before submit
SEND_VARIATION	Export	B	“B” means that the system should tolerate error messages on submit
WITHDRAW_PROCESS_BUTTON_VISIBLE	Export	ALWAYS	Allows the possibility of cancelling the ticket on this step
PROCSTATE	Import		Process Status after step execution

Table 17 – SAP Workflow Step – Edit & Submit Form Data

As configured in the workflow’s basic data, the program exit *CL_HRASR00_POBJ_WF_EXIT* is also attached to this step. That will allow the automatic creation of a step object (similar to the process object, but which will contain form data submitted in this particular step).

5.4.2.3 Import Ticket Effective Date

This step runs once the form is submitted into SAP. Its purpose is similar to “Import Ticket Identifier” step, but in this case, it is fetching form field “Effective Date”, which holds the day these personnel changes take effect. The same standard task is used, and the binding is:

Field Name	Type	Value	Description
FORM	Export	YP01_PCX 0000	Form Scenario (identifies the process/form)
FORM_FIELD_NAME_1	Export	EFFECTIVE_DATE	Field name to be read
FORM_FIELD_VALUE_1	Import		Effective Date

Table 18 – SAP Workflow Step – Import Ticket Effective Date

This date is relevant for the Assisted Posting phase.

5.4.2.4 Condition “Is Manual Post”

This is a condition step without any task involved. It validates the possibility of executing the Assisted Posting functionality. For example, legacy tickets (from the old version of the application) should not allow the use of assisted posting, because all data was already loaded manually into SAP. This is controlled by the container element *LOADED_TO_SAP*, which is set by the API. Figure 56 shows the condition as it was set.

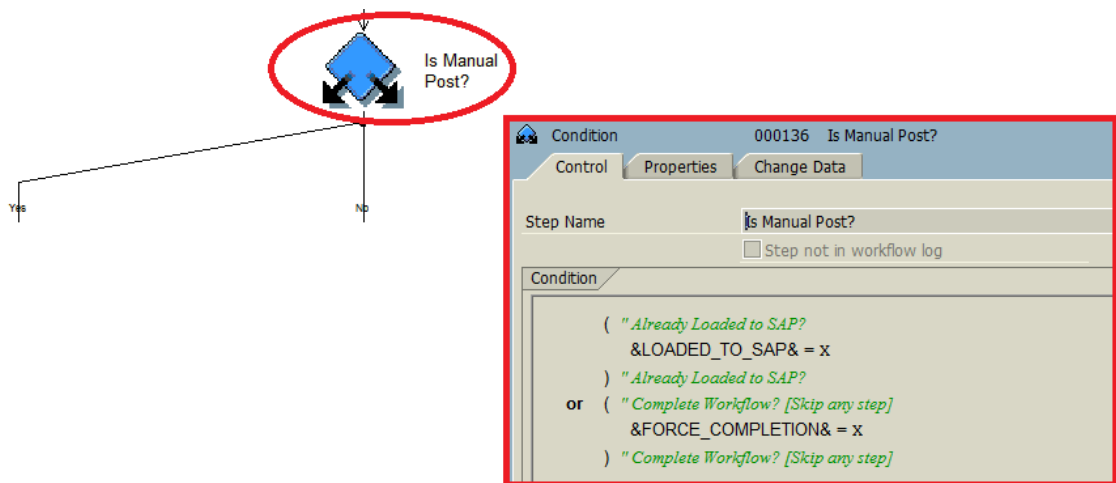


Figure 56 – SAP Workflow Step – Condition “Is Manual Post”

5.4.2.5 SAP Assisted Posting

Assisted Posting is a phase where the SAP Administrator is responsible for processing the ticket within the SAP ERP. This part of the process consists of several background steps and an interactive step that starts in the SAP Business Workplace under the form of a work item. “Work items are used in the SAP Business Workflow/WebFlow to execute the individual steps of a workflow” (SAP SE, 2016s).

Once this work item is executed, the system launches the backend console, which is the interface/program that provides semi-automated functions to execute the necessary personnel data changes.

Figure 57 gives an overview of the process in SAP ERP.

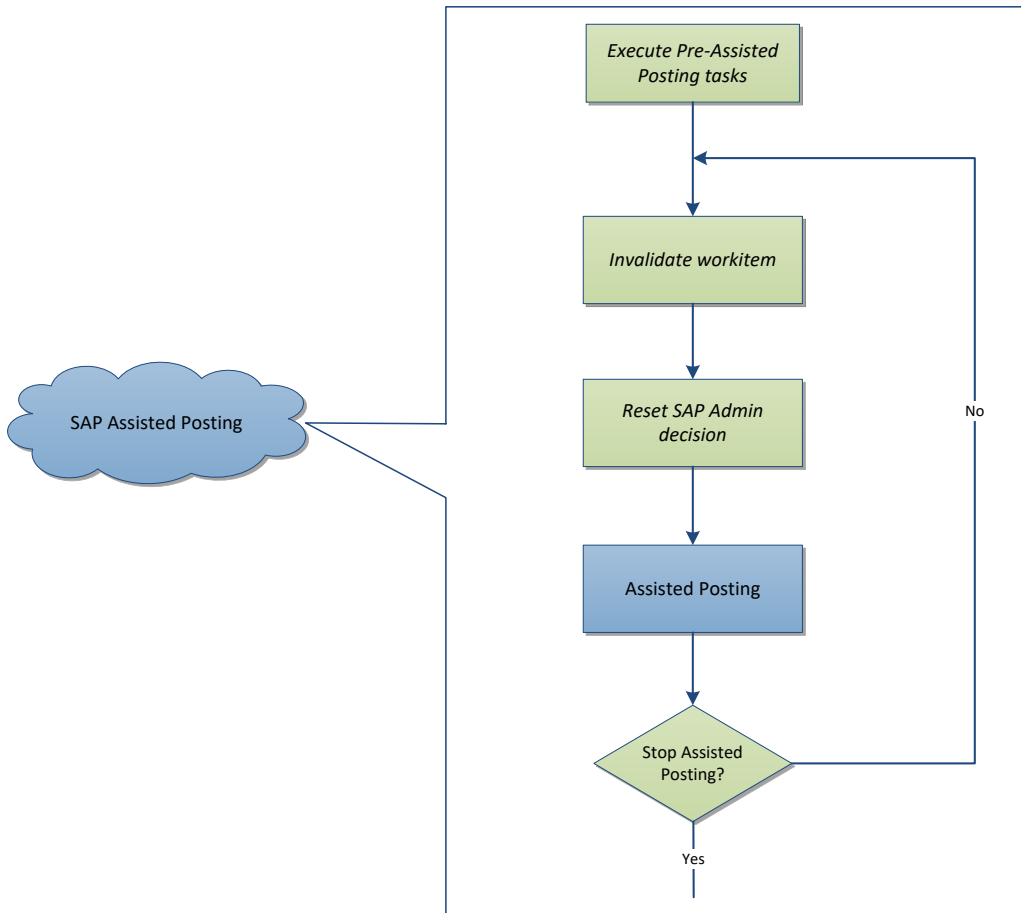


Figure 57 – SAP Workflow – Assisted Posting

5.4.2.5.1 Execute Pre-Assisted Posting tasks

The first step is a background task responsible for preparing the assisted posting. Technically, this step is bonded to a custom task, created specifically for this purpose: *TS97300010 – ePCX Pre Assisted Posting Tasks*. In turn, this task's logic is isolated in the utility class previously presented. This task is therefore a mirror of an ABAP method.

The actions executed by this task/method are:

- Save of ticket data to be loaded in SAP infotypes;
- Preparation of the Business Workplace and Backend Console.

The mapping between the workflow and this step is as presented in Table 19. Note the two parameters being imported, used to build the Business Workplace and Backend Console:

Field Name	Type	Description
PCX_ID	Export	Ticket Identifier
EFFECTIVE_DATE	Export	Effective Date

Field Name	Type	Description
S_BWP_DATA	Import	Ticket summary to be displayed in Business Workplace
IT_ASSISTED_POSTING_TASKS	Import	List of tasks that will be made available to the user inside the Backend Console

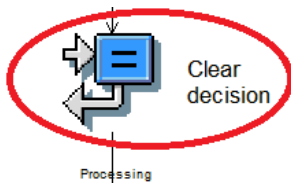
Table 19 – SAP Workflow Step – Execute Pre-Assisted Posting tasks

5.4.2.5.2 Invalidate Work Item

In each loop iteration, the previous work item must be marked as no longer valid, so that it is not used again by the API. This step is described in detail on section 5.4.2.9 Invalidate Work Item.

5.4.2.5.3 Reset SAP Admin Decision

Since the SAP Administrator can reject tickets (and therefore, the process can enter into a loop) it is necessary to have a container element that stores the last decision made by the SAP Administrator. This decision must be reset before making the work item available to the user. To accomplish this, container element *SAP_ADMIN_DECISION* is initialized with the value *PROCESS*, using a step of type “Container Operation”, as displayed in Figure 58:



The screenshot shows the configuration for a 'Container operation' step named 'Clear decision' with ID '000125'. The 'Control' tab is active. The 'Step Name' is 'Clear decision' and the 'Outcome name' is 'Processing'. There is a checkbox for 'Step not in workflow log' which is checked. The 'Operation' section shows: 'Result Element' is 'SAP_ADMIN_DECISION', 'Assignment' is '=', and the expression is 'PROCESS'. The operator is 'Assignment'.

Figure 58 – SAP Workflow Step – Reset SAP Admin Decision

5.4.2.5.4 Assisted Posting

This workflow step is the main point of entry to process the ticket in SAP. Technically, this step launches an SAP program (*Y_PCX_BACKEND_CONSOLE*) which won't be detailed as it is not the

focus of this document. However, it's important to know that this program is the central tool to process the personnel data changes. It provides the user with automatic and semi-automatic steps to proceed with infotype maintenance and reconciliation of data. This step is bonded to a custom task (*TS97300015 – ePCX Assisted Posting*), which in turn is connected with the program described above. Besides that, it has similarities with the step “Edit & Submit Form Data”, because the ticket may be edited, approved or rejected.

The binding between the workflow and this step is configured as displayed in Table 20:

Field Name	Type	Value	Description
PCX_ID	Export		Ticket Identifier
EFFECTIVE_DATE	Export		Effective Date
S_BWP_DATA	Export		Ticket summary to be displayed in Business Workplace
SCREEN	Export	<i>(blank)</i>	Controls if the Report's selection screen should appear ('X') or should be skipped (<i>blank</i>)
FORM	Export	YP01_PCX 0000	Form Scenario (identifies the process/form)
FORM_SCENARIO_STAGE	Export	YP01_PCX_A	Form Scenario Step
SAVE_DRAFT_BUTTON_VISIBLE	Export	X	Allows the possibility of a draft before submit
PROCSTATE	Import		Process Status after step execution

Table 20 – SAP Workflow Step – Assisted Posting

Program Exit *CL_HRASR00_POBJ_WF_EXIT* is also assigned to this step so that the synchronization between the workflow and the Process Object is maintained (similarly to other interactive tasks). Furthermore, the agent responsible for this step is given by the container element *SAP_ADMIN*, which is set by the API when the ticket is pushed to SAP.

5.4.2.5.5 Condition “Stop Assisted Posting”

This loop condition validates if the SAP Administrator has already made a decision regarding the ticket. Therefore, if the *SAP_ADMIN_DECISION* is *CLOSE / REVERT* or the ticket is pulled back from the web application (HCM P&F event “*BACK*”), the workflow will exit assisted posting. Note that this condition does not decide what the next step should be; that will be seen later. Here, we are only interested in checking if the assisted posting phase has ended.

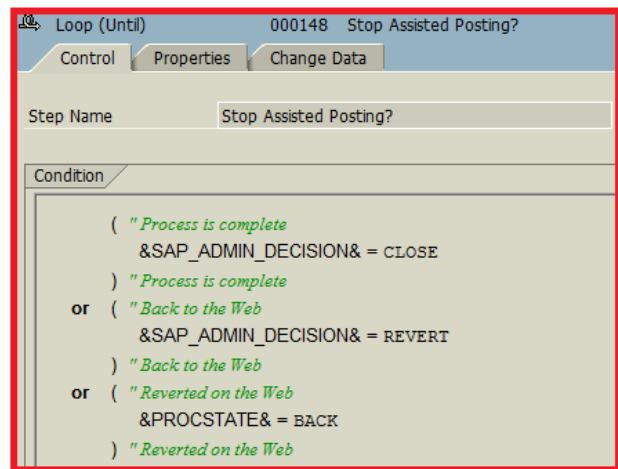
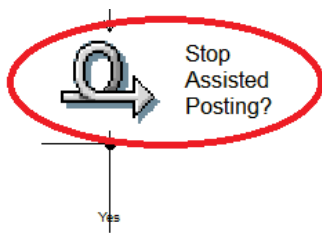


Figure 59 – SAP Workflow Step – Condition “Stop Assisted Posting”

5.4.2.6 Bypass Assisted Posting

This step is only relevant if the result of condition “is manual post” was true, meaning that the ticket isn’t meant to be processed through assisted posting. Technically, this step just forces the workflow to move forward even without having a decision from the SAP Administrator. That is accomplished by setting the field *SAP_ADMIN_DECISION* to *CLOSED*. There is no workflow task involved, just a standard workflow step that allows the change of workflow data.

5.4.2.7 Condition “Close Assisted Posting”

This condition checks if the SAP phase is complete. In other words, this condition will evaluate the decision taken by the SAP Administrator (container element *SAP_ADMIN_DECISION*). The possible actions are:

- Close the SAP posting phase:
 - Move on to “Review Form Data (Manual Step)” or complete the workflow.
- Send back the ticket:
 - Move back to “Edit & Submit Form Data”.

Figure 60 shows the condition applied by this step.

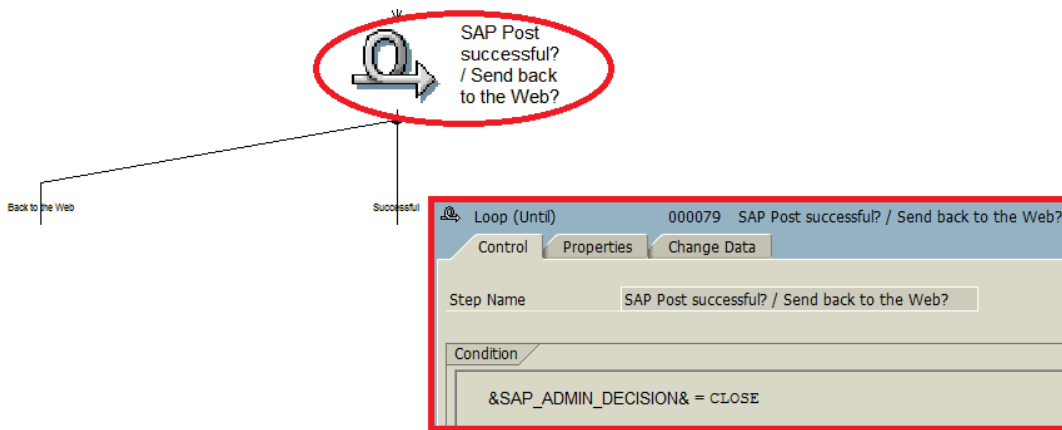


Figure 60 – SAP Workflow Step – Condition “Close Assisted Posting”

5.4.2.8 Condition “Complete Workflow”

This condition checks if the “Review Form Data (Manual Step)” step should be part of the process. If the container element *FORCE_COMPLETION* is true, the workflow will end immediately.

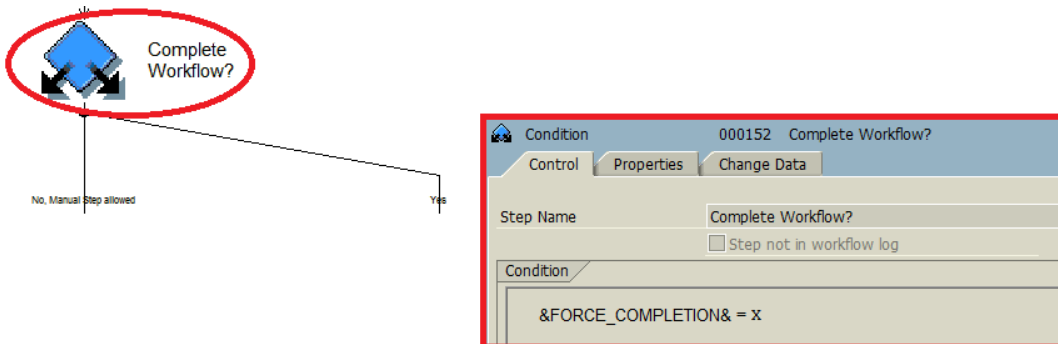


Figure 61 – SAP Workflow Step – Condition “Complete Workflow”

5.4.2.9 Invalidate Work Item

After the conclusion of an interactive step, it’s necessary to invalidate the work item being used for communication between the web application and SAP ERP. This is required due to the fact that the work item is no longer valid for reading/saving data in the Process Object. The new work item should be acknowledged once communication is restarted.

To hold this logic, task *TS97300013 – ePCX Discard process work item* was created as a wrapper to an ABAP method within the utility class, which only requires the ticket identifier.

Field Name	Type	Description
PCX_ID	Export	Ticket Identifier

Table 21 – SAP Workflow Step – Invalidate Work Item

5.4.2.10 Revert SAP Post

This is a background step. It is only executed if the SAP Administrator chooses to send the ticket back and its main goal is to communicate that info to the web application. The workflow will then move to “Edit & Submit Form Data” step (i.e., return to the starting point). The binding between the workflow and this step is as displayed in Table 22:

Field Name	Type	Description
PCX_ID	Export	Ticket Identifier
REVERT_COMMENTS	Export	SAP Admin Comments

Table 22 – SAP Workflow Step – Revert SAP Post

The communication between SAP ERP and the web application is done through a simple Web Service that was created specifically for this purpose. Task *TS97300017 – ePCX Revert SAP Post* is responsible for this communication. Once more, this task delegates that in a method of the utility class.

5.4.2.11 Review Form Data (Manual Step)

The workflow last step is similar to “Edit & Submit Form Data”. It enables users to edit the form and it is also connected to standard task *TS17900100 – Process Form*.

This step was created to allow form edition during payroll phase, or in case the payroll administrator detects some inconsistency that needs to be manually fixed (Post to SAP Manual).

5.5 P&F External Events

5.5.1 Technical Overview

As previously pointed out, some events could not be configured within HCM P&F, either because they didn’t support customer requirements or due to performance issues. For this reason, an infrastructure with similarities to HCM P&F was put in place. The primary necessity was that it remained a dynamic and configurable application, so the components were developed with that goal in mind.

This infrastructure is divided in two components:

- A configuration table – *YPCX_CUST_EVENTS* – to hold custom events, their fields and a reference to the logic that should run for each one;
- A class – *YCL_PCX_CUSTOM_EVENTS* – that holds each event’s logic.

The integration between these two components is then handled by the API, which reads the configuration table, finds the event and the reference to a class method in order to dynamically execute it:

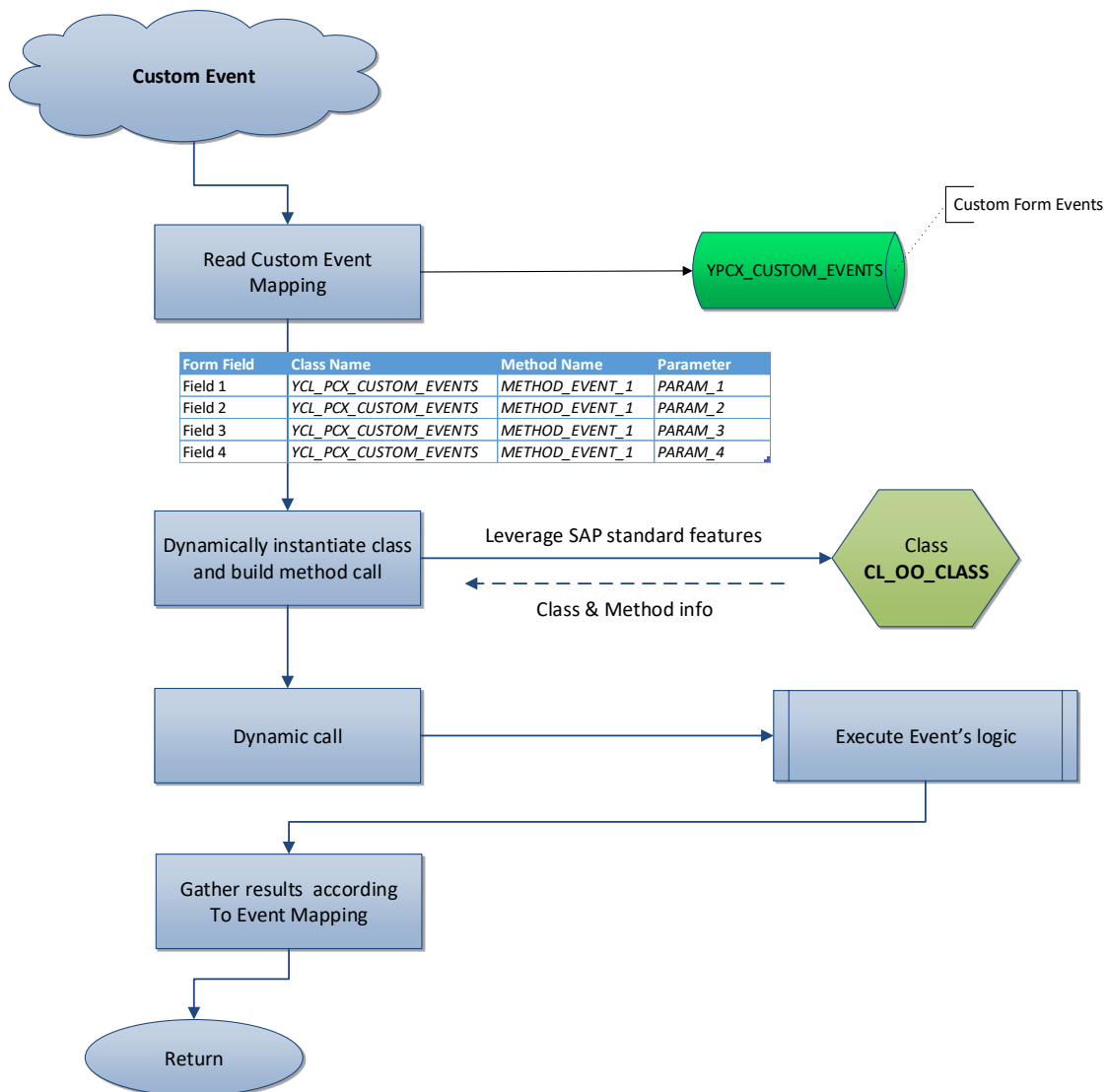


Figure 62 – P&F External Events – Flow of Execution

Table 23 shows the structure of table *YPCX_CUST_EVENTS* – a raw structure with all the necessary data to execute an event:

Field	Key	Data Type	Description
MANDT	X	MANDT	Client
FORM_SCENARIO	X	ASR_FORM_SCENARIO	Form Scenario
EVENT	X	ASR_FORM_OPERATION	Event Name
SEQNR	X	NUM03	Sequence Number
CLASS		SEOCLSNAME	Name of the class that holds the logic

Field	Key	Data Type	Description
METHOD		SEOCMPNAME	Name of the method that holds the logic
FIELD_NAME		ASR_FIELDNAME	Fieldname
FIELD_TYPE		YPCX_EVT_FIELD_TYPE	Field Type (Import/Export)
INPUT_PARAMETER		SEOSCONAME	Method Parameter Name
INPUT_PARAM_TYPE		RS38L_TYP	Method Parameter Type
INPUT_PARAM_FLD		FIELDNAME	Parameter fieldname (for structures)
SUFFIX		YPCX_CUSTOM_EVT_SUFFIX	Suffix (for structures)
DESCRIPTION		ASR_FIELDNAME_TEXT	Suffix description (for structures)
SORTING		YPCX_EVT_PARAM_SORTING	Results Sorting
GROUPING_ID		FIELDNAME	Grouping identifier (pattern for results display)
GROUPING_ORDER		INT2	Grouping order (pattern for results display)
GROUPING_PATTERN		YPCX_GROUPING_PATTERN	Grouping pattern (pattern for results display)

Table 23 – YPCX_CUST_EVENTS – Structure

Most of the events are related to Organizational Management data, like organizational units and positions, objects particularly sensitive due to the large amount of data in the system.

5.5.2 Event Types

The custom events that were created can be divided in four groups:

- Check values
- Default values
- Basic Search
- Full Search

Basic Search corresponds to the recalculation of possible values, based on the changes made to other form fields (similar to HCM P&F). In addition, *Full Search* events allow users to select the filters they please from a list, in order to look for the accurate value. In the web application, this

is a modal that opens on user demand. Figure 63 is an example of the full search functionality available for Organizational Units.

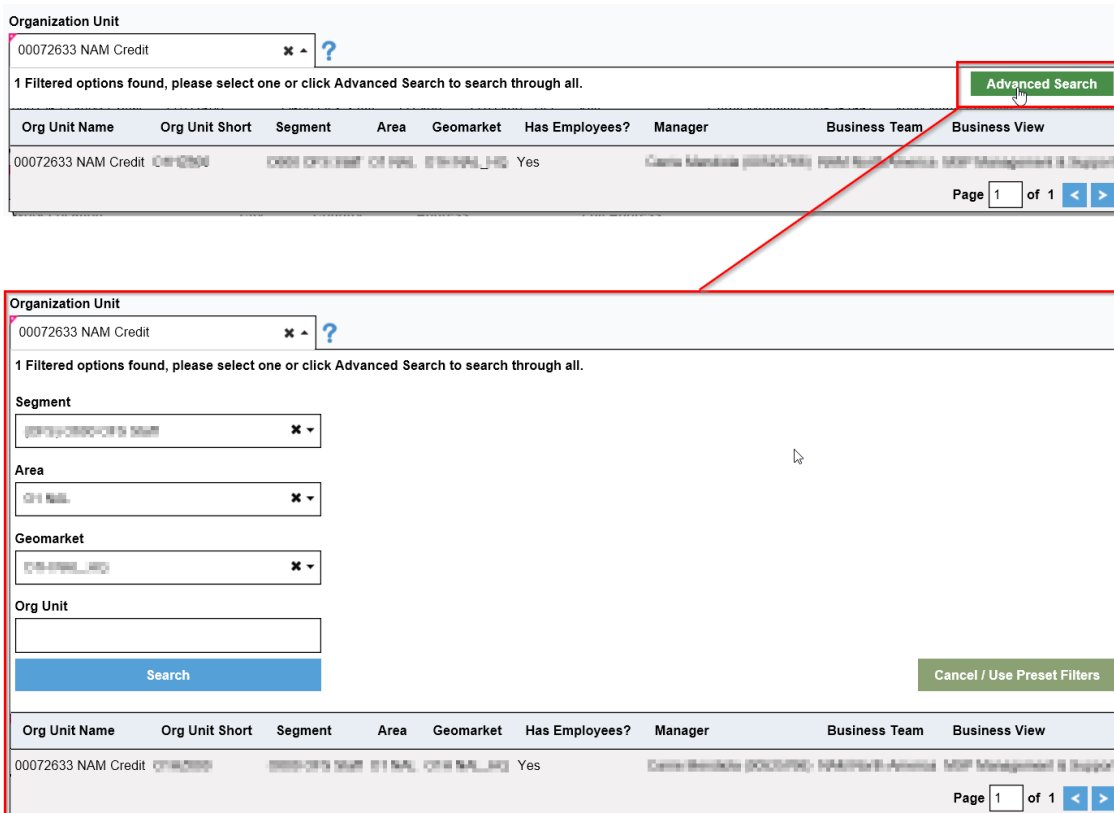


Figure 63 – Full Search – Organizational Unit

Basic Search events were implemented for:

- Direct Manager
- Organizational Unit
- Cost Center
- Work Center
- Job and their family of fields
- Position Title
- Personnel Area & Company Code
- Payscale Group & Level
- Work Schedule Rule

Full Search events were implemented for:

- Organizational Unit
- Cost Center
- Job
- Work Center
- Personnel Area

5.6 Code Review

Considering the dimension of this project, one stage of evaluation for the solution was the use of code audit functionalities to assure the code respects good practices. SAP SE provides an analysis tool called Code Inspector which evaluates ABAP objects and helps developers / quality managers to follow good practices by detecting issues and suggesting possible optimizations for the code (SAP SE, 2016u).

Code Inspector is also customizable, so new rules can be applied (SAP SE, 2016u) if there are some specificities not covered by SAP standard. For the purpose of this project, only standard rules were used. These rules are divided in four categories:

- Performance Checks
- Security Checks
- Syntax Check/Generation
- User Interfaces

Messages				Error	Warnings	Information	
	D...	...	E...	Tests			
				List of Checks	2	13	0
▶				Performance Checks	0	13	0
▶				Security Checks	0	0	0
▶				Syntax Check/Generation	2	0	0
▶				User Interfaces	0	0	0

Figure 64 – Code Inspector Categories

As an example, let's consider a *SELECT* statement that does not fall into any database index (either primary or secondary). Code Inspector detects this issue and suggests developers to review the code and/or database indexes in order to improve performance:

Messages				Error	Warnings	Inform...	
	D...	...	E...	Tests			
				List of Checks	0	7	0
▶				Performance Checks	0	7	0
▶				Analysis of WHERE Condition for SELECT	0	3	0
▶				Warnings	0	3	0
▶				Message Code 0003 ==> Table YPCX_PAREA: No First Field of a Table Index in WHERE Condition	0	2	0
▶				Message Code 0001	0	1	0
▶				Analysis of WHERE Condition in WHERE and Omit	0	1	0
▶				WHERE statements that bypass the Table Buffer	0	1	0
▶				Low-Performance Operators on Internal Tables	0	1	0
▶				Localities, Recursive Operations	0	1	0
▶				Indices Creation of Views	0	1	0
▶				Table Structure Checks	0	1	0
▶				Security Checks	0	1	0
▶				Syntax/Check/Generation	0	1	0
▶				User Interfaces	0	1	0

Figure 65 – Code Inspector Example

Code Inspector was frequently used during developments and always before considering functionalities ready for test. This helped tackle performance/security issues right from the start, so although it was a minor point in the project timeline, it was surely an important step during implementation phase.

6 Evaluation

Considering the amount of users that will use this application as well as the specificities of each country involved, several mechanisms were used to evaluate the solution.

The principal points of evaluation were:

- User Satisfaction
- Performance / Response Time
- SAP System Usage (CPU %, etc.)

The solution passed through two test phases: alpha and beta. Alpha tests were focused on functionalities, whereas beta tests were done by real users – people with business knowledge – and therefore, focused on business scenarios.

Regarding performance/response time evaluation, the test scenarios were ticket creation and the most complex form events (triggered by user action). There were two evaluation phases: one that started before alpha tests, where two samples were compared (before/after performance tuning) and one final evaluation post alpha tests, with just one sample, comparing it to what is considered a reasonable result. Additionally, system usage was evaluated through load tests.

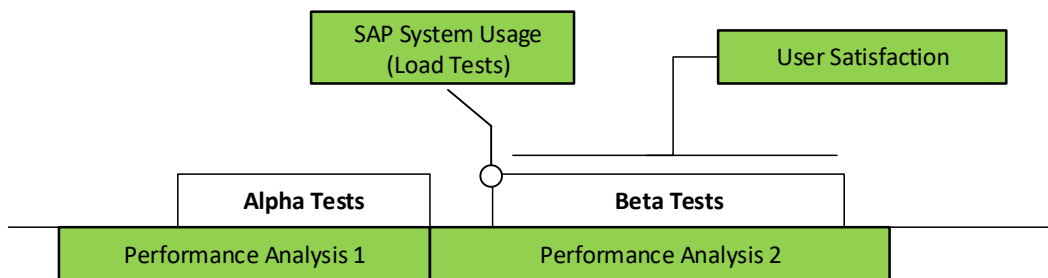


Figure 66 – Evaluation Timeline

6.1 User Satisfaction

In order to evaluate user satisfaction, test scripts were provided to beta test users. Using these scripts, users were asked to signal the successful/unsuccessful steps, paste screenshots of errors and provide their input on improvements or new features. This obviously involved applying fixes and re-test, but here we'll just focus on determining if the ratio errors–enhancements differ from what is expected. Note that previous alpha test errors were not accounted for this, as well as user errors or errors that were soon established as non-issues.

It was defined as expected errors: 40 and expected enhancements: 10. This was based on the fact that the beta tests took 2 weeks, with 4-5 people using the application each day. The results are displayed in Table 24.

	Errors	Enhancements / Incomplete Specifications	Total
Observed	44	17	61
Expected	40	10	50

Table 24 – User Satisfaction – Observed vs. Expected

It's possible to see that there were 22% more items than what was expected and the enhancements requested almost doubled (70% increase). These numbers were analyzed and it was concluded that users were mentioning features from the old application, earlier discarded by the company, during requirements phase. Further discussions took place in order to define what would be implemented or not post go-live.

To summarize, apart from that, this sample of beta test users were satisfied with the application. Later on section 7.2 Critical Reception, under 7 Summary, a vaster user review will be addressed.

6.2 Performance / Response Time

6.2.1 Phase 1

The first phase of performance evaluation started before alpha tests. Several tests were made to the most complex form events and results compiled. From these results, it was already possible to determine events that would require some tuning.

In order to understand where the issues were placed and check tips for optimization, two SAP tools were used: transaction *SAT*, SAP technology to analyze runtime execution (Dolinskaja, 2011) and *ST05*, the most important tool to test database performance through SQL traces (Boes, 2007).

Transaction *SAT* allows developers to measure programs/functions execution, by giving the execution time per instruction. So, it's possible to identify methods taking much of the execution or go even deeper into ABAP calculations, database accesses, etc.

In Figure 67 and Figure 68, it's possible to see part of an *SAT* measurement done to *Y_PCX_PROCESS_EVENT*, which took around 0,7 seconds.

Measurement: Display

Trace Display for: Y_PCX_PROCESS_EVENT

Date/Time: 24.11.2014 12:44:21

Desktop 1 Desktop 2 Hit List Prof. Times Mod. Tree Call Hierarchy

Profile	Number	Time in Microsec.	1/1000 Portn.
Object Profile		681.084	
Datenbank-Operationen	6541	222.863	327
Buffer Operations	1899	5.739	8
SAP SQL	2980	203.076	298
Export/Import Database	25	1.500	2
DB-Operations	384	9.559	14
Exporting/Importing	1253	2.989	4
Modularisierungseinheiten	24836	444.665	653
Form-Routinen	11498	175.637	258
Function Calls	5607	120.260	177
Methoden/Events	7731	148.768	218
Sonstiges	1809	7.876	12
Dynpro/Transaction Control	1337	3.248	5
LOAD	253	2.280	3
Zeitmessung	2	1.267	2
Various ABAP-Instructions	217	1.081	2

Figure 67 – Transaction SAT – Example

Measurement: Display

Trace Display for: Y_PCX_PROCESS_EVENT

Date/Time: 24.11.2014 12:44:21

Desktop 1 Desktop 2 Hit List Prof. Times Mod. Tree Call Hierarchy

Legend Goto Tot.Up Tot.Down Over.View Call Stack ABAP

Call Hierarchy

Index	Level	Hierarchy Field	Gross	Net	AddField	Index
1	0	>Runtime analysis	679.542	3		0
2	1	>_Runtime analysis	679.539	1.264		0
3	2	>_Load Report SAPLYPCX	16	16		0
4	2	<_Load Report SAPLYPCX	0	0		0
5	2	>_Call Func. Y_PCX_PROCESS_EVENT	678.259	293		0
6	3	>_Load Report YCL_PCX_SAP_API=====CP	13	13		0
7	3	<_Load Report YCL_PCX_SAP_API=====CP	0	0		0
8	3	>_Call M. YCL_PCX_SAP_API=>GET_INSTANCE	2.074	16		0
9	4	>_YCL_PCX_SAP_API	3	3		0
10	4	<_YCL_PCX_SAP_API	0	0		0
11	4	>_Call M. 15<YCL_PCX_SAP_API->>CONSTRUCTOR	2.055	118		0
12	5	>_Call M. 15<YCL_PCX_SAP_API->>GET_MESSAGE_HANDLER_INSTANCE	33	21		0
13	6	>_Load Report YCL_MESSAGE_HANDLER=====CP	3	3		0
14	6	<_Load Report YCL_MESSAGE_HANDLER=====CP	0	0		0
15	6	>_YCL_MESSAGE_HANDLER	3	3		0
16	6	<_YCL_MESSAGE_HANDLER	0	0		0
17	6	>_Call M. 16<YCL_MESSAGE_HANDLER->>CONSTRUCTOR	6	3		0
18	7	>_Call M. 16<YCL_MESSAGE_HANDLER->>ENABLE_CONTROL_DUPLICATED_MSGS	3	3		0
19	7	<_Call M. 16<YCL_MESSAGE_HANDLER->>ENABLE_CONTROL_DUPLICATED_MSGS	0	0		0
20	6	<_Call M. 16<YCL_MESSAGE_HANDLER->>CONSTRUCTOR	0	0		0
21	5	<_Call M. 15<YCL_PCX_SAP_API->>GET_MESSAGE_HANDLER_INSTANCE	0	0		0
22	5	>_Load Report YCL_PCX_UTILS=====CP	2	2		0

Figure 68 – Transaction SAT – Example – Time per instruction

For scenarios where the execution exceeded a reasonable time in database accesses, transaction *S705* was used. This transaction allows developers to activate several type of traces.

For the purpose of evaluating this product, SQL trace was activated so that every database call was registered. One can then navigate through those trace results and (Boes, 2007):

- Display the number of database accesses / table details;
- Display SQL statement details (values used, number of records);
- Check execution time per SQL statement, divided in *PREPARE*, *OPEN*, *FETCH*, etc.;
- Execute “Explain” function to find out how the statement was processed, particularly which index was used.

The following example shows a trace done to the “Cost Centers Full Search” event, which accepts up to five filters.

Test for function group YPCX
 Function module Y_PCX_PROCESS_EVENT
 Uppercase/Lowercase

Runtime: 1,432,006 Microseconds

RFC target sys:

Import parameters	Value
IV_PCX_ID	USC2BXCFFJNVU
IT_EVENTS	1 Entry
IT_DATA	5 Entries

Export parameters	Value
ET_DATA	0 Entries
ET_ADDITIONAL_DATA	54 Entries
ET_SUGGESTED_DATA	0 Entries
ET_MESSAGES	0 Entries
EV_IS_OK	X

Figure 69 – Event – Cost Center Full Search

SQL Trace was activated before execution:

Performance Analysis

Select Trace

- SQL Trace
- Enqueue Trace
- RFC Trace
- Table Buffer Trace

Select Trace Function

- Activate Trace
- Activate Trace with Filter
- Deactivate Trace
- Display Trace
- Enter SQL Statement

Figure 70 – ST05 – Activate SQL Trace

Cost Center search is based on a big table (~3M entries) in SAP ERP. Looking at the results, we can find and analyze the instruction(s) that access this table:

Statement							
DDIC information Explain							
Transaction SE37		Work process no 19		Proc. Type DIA	Client 300	User BATCHAPP03	TransGUID 5805A3558D6B2B89E1000
Duration	Obj. name	Op.	Recs.	RC	Statement		
133,333	SE03POCO	FETCH	104	0			
23,333	SE03POCO	OPEN	0	1803			
11	SE03POCO	RELEASE		0	SELECT WHERE TOLANGD='-JA' AND TRESIDID='-JA'		
111	SE03POCO	RELEASE		0	SELECT WHERE TOLANGD='-JA' AND TRESIDID='-JA'		
1	SE03POCO	OPEN		0	SELECT WHERE TOLANGD='-POLPOL_CONTROLLER' AND TRESIDID='0'		
177,333	SE03POCO	FETCH	187	0			
277,333	SE03POCO	OPEN	287	0			
18,211	SE03POCO	OPEN	0	1803			
11	SE03POCO	DECLARE		0	SELECT WHERE TOLANGD='-JA' AND TRESIDID='-JA'		
116	SE03POCO	RELEASE		0	SELECT WHERE TOLANGD='-JA' AND TRESIDID='-JA'		
1	SE03POCO	OPEN		0	SELECT WHERE TOLANGD='-POLPOL_CONTROLLER' AND TRESIDID='1'		
1,544	SE03POCO	OPEN	0	1803			
1	TRFC_DTT	RELEASE		0	SELECT WHERE TRANDT='1300' AND TRESCHNR='TRFC_PROCESSOR'		
199	TRFC_DTT	RELEASE	1	0			
1	TRFC_DTT	RELEASE		0	SELECT WHERE TRANDT='1300' AND TRESCHNR='TRFC_PROCESSOR'		
187	TRFC_DTT	RELEASE	1	0			
1	TRFC_DTT	RELEASE		0	SELECT WHERE TRANDT='1300' AND TRESCHNR='TRFC_PROCESSOR'		
11	TRFC_DTT	RELEASE	1	0			
21	(DECLARE		0	SELECT WHERE mandt = '300' AND processed = 'X' AND begda <= '20161017' AND		
143	(PREPARE		0	SELECT WHERE mandt = '300' AND processed = 'X' AND begda <= '20161017' AND		
5	(OPEN		0	SELECT WHERE mandt = '300' AND processed = 'X' AND begda <= '20161017' AND		
327,450	(FETCH	6	1403			
1	TRFC_DTT	RELEASE		0	SELECT WHERE TRANDT='1300' AND TRESCHNR='TRFC_PROCESSOR'		
104	TRFC_DTT	RELEASE	1	0			
1	TRFC_DTT	RELEASE		0	SELECT WHERE TRANDT='1300' AND TRESCHNR='TRFC_PROCESSOR'		
181	TRFC_DTT	RELEASE	1	0			
1	TRFC_DTT	RELEASE		0	SELECT WHERE TRESCHNR='TRFC_PROCESSOR' AND TRESIDID='0' AND TRESIDID='TRFC_PROCESSOR'		
181	TRFC_DTT	RELEASE	1	0			
181	TRFC_DTT	RELEASE	1	0			
181	TRFC_DTT	RELEASE	1	0			

Figure 71 – ST05 – Cost Center Full Search Results

After gathering information from SAT and ST05, we can start looking for possible optimizations, following good practices provided by SAP SE in *Enhancing the Quality of ABAP Development* (Heuvelmans, et al., 2004).

These principles were applied to several custom events, plus the ticket creation, as already stated. These were some of the tips used to tune these events:

- Avoid databases accesses inside loops;
- Reduce database accesses by using *INNER JOIN / SELECT IN*;
- Minimize the time of open connection between ABAP server – Database server and fetch only the required data;
- Shift logic from database into ABAP servers (for example, avoid *ORDER BY*; simply return the results and sort them in ABAP);
- Assure the correct index is used or create news when strictly necessary;
- Work with buffered tables if required;
- Use binary searches and parallel cursor method for nested loops in order to increase ABAP performance.
- etc.

As an example, let's look at the results of “Work Center Basic Search” event (before and after performance tuning). Although this is not one of the most problematic events, it's a good example of what's possible to do even for events taking around 2 seconds.

These were the samples used for comparison, taken on different days, but for the same scenarios:

Test	Before (sec)	After (sec)	Difference (sec)
1	2,39	0,64	1,756
2	0,47	0,34	0,132
3	1,79	0,29	1,499
4	3,10	0,31	2,789
5	3,00	0,30	2,697
6	0,89	0,34	0,547
7	2,11	0,31	1,799
8	0,61	0,30	0,31
9	2,04	0,31	1,734
10	1,05	0,34	0,708
11	0,48	0,32	0,155
12	1,38	0,29	1,088
13	0,47	0,31	0,164
14	0,90	0,31	0,59
15	0,96	0,46	0,501
16	2,19	0,53	1,665
17	1,97	0,48	1,485
18	1,29	0,30	0,994
19	1,17	0,30	0,872
20	2,48	0,37	2,109

Table 25 – Work Center Basic Search – Execution Time Results

From the box plot in Figure 72, we confirm that the data is almost symmetric and proceed with the evaluation, using a paired-sample t test.

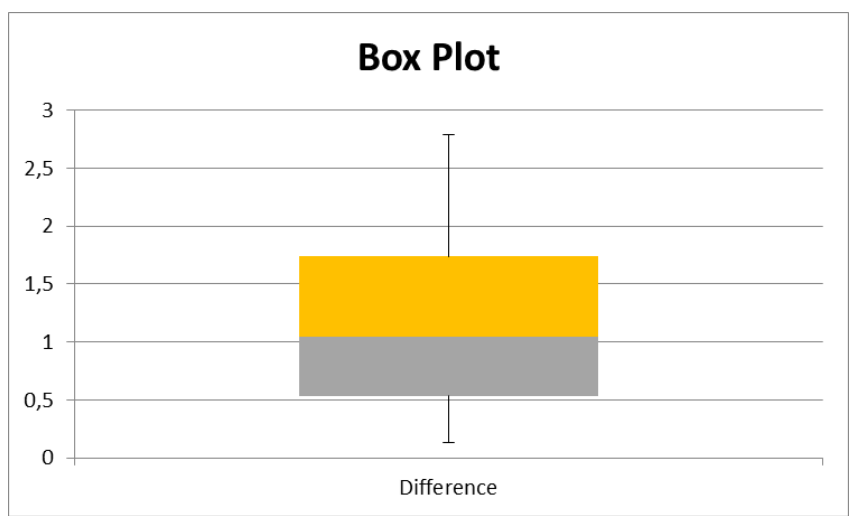


Figure 72 – Work Center Full Search – Box Plot

These were the results from the test:

SUMMARY		Alpha		0,05		Hyp Mean Diff		0	
Groups	Count	Mean	Std Dev	Std Err	t	df	Cohen d	Effect r	
Before	20	1,54	0,837134						
After	20	0,36	0,094303						
Difference	20	1,1797	0,818011	0,182913	6,449521	19	1,442157	0,828523	

T TEST					
	p-value	t-crit	lower	upper	sig
One Tail	1,75152E-06	1,729133			yes
Two Tail	3,50303E-06	2,093024	0,796859	1,562541	yes

The mean reduced from 1,54 seconds to 0,36 seconds, which is a 76,7% decrease in runtime execution. As displayed in column *sig*, since $p\text{-value} = 3,50303E-06 < \alpha = 0,05$, it's possible to conclude that these results are reliable and there was a major reduction in execution time.

6.2.2 Phase 2

The second phase of performance evaluation took place entirely in the quality system, where there is a lot more production-like data; fact that increases the credibility of these tests.

At this point, it was defined the acceptable response time of each event in conversations with the customer (this is basically a comparison against the old application). Note that this was a continuous effort that lasted through beta tests. As an example, let's look at ticket creation. The customer defined 12 seconds as the average acceptable time. The samples extracted from the tests were as follows:

Test	Execution Time (sec)
1	7,95
2	9,94
3	7,51
4	6,31
5	9,04
6	9,63
7	10,56
8	10,89
9	7,89
10	15,79
11	8,92
12	9,77
13	8,63
14	8,12

Test	Execution Time (sec)
15	11,12
16	9,22
17	9,01
18	7,01
19	9,79
20	8,31

Table 26 – Ticket creation – Execution Time Results

From the box plot in Figure 73, we see that the data is fairly symmetric so it's correct to use *t* test, even though the sample is small.



Figure 73 – Ticket creation – Box Plot

The null hypothesis here is defined by the average set by the customer:

$$H_0: \mu = 12$$

Using a one sample *t*-test, we get the following results:

SUMMARY			Alpha	0,05			
Count	Mean	Std Dev	Std Err	t	df	Cohen d	Effect r
20	9,27	1,984569	0,443763	-6,15149	19	1,375515	0,815925

T TEST		Hyp Mean		12		
	p-value	t-crit	lower	upper	sig	
One Tail	3,26E-06	1,729133			yes	
Two Tail	6,52E-06	2,093024	8,341388	10,199	yes	

It stands out a mean of 9,27 (below 12) and since $p\text{-value} = 3,26E-06 < \alpha = 0,05$, it's possible to conclude that there is a significant time reduction from the average established by the customer.

All events causing issues were assessed with this principle; fact that, in the end, would assure good performance / response time of the whole application.

6.3 SAP System Usage

In order to complement performance evaluation, the SAP system behavior/response had to be measured during unusual peak load conditions. The purpose of this is to assure that the operating capacity of the whole application is meeting what was specified and/or identify possible bottlenecks, modes of failure, etc.

To accomplish this, load tests were done to ticket creation and some form events. In total, these load tests took approximately an hour and a half. During those 90 minutes, several calls were made to the SAP system, including the creation of nearly 500 new tickets. To help analyze the results of these tests, the following SAP transactions were used:

- *ST03N* (Workload Monitor) – As the name indicates, this transaction displays data about the workload that is written by SAP kernel. It provides various details, such as, number of calls, average/absolute response times, CPU time and DB accesses (SAP SE, 2016u).
- *ST06* (Operating System Monitor) – This transaction was used to check CPU utilization, max and average.
- *SM21* (System logs) – This transaction shows logs created locally in all instances of the SAP system (SAP SE, 2016x). This was helpful to check warnings/errors that could be produced by HCM P&F.

During the load tests, a total of 17.708 HTTP calls were made, shared by two application servers. Furthermore, *ST06* didn't show any outstanding issues – there was an average CPU utilization of 50% in one application server and 22% in the other. Response time (both CPU and DB) were also checked in order to identify critical events that would require some tuning (already presented under 6.2 Performance / Response Time). High CPU time would usually indicate potential for ABAP tuning, whereas high DB time would be related to DB accesses. An overview of the results can be seen in Table 27.

Statistic Name	Value
Number of Calls	17.078
Total Call Time (s)	13.523
Total Data Sent (Bytes)	839.262.515
Total Data Received (Bytes)	35.986.842
Average Call Time (ms)	791,9
Average Quantity of Data Sent (Bytes)	49.142,90
Avg. Recd Data Quantity (Bytes)	2.107,20

Table 27 – SAP System Usage – Overview

7 Summary

This chapter contains the final results of the work behind this dissertation, including user's reception and possible future work.

7.1 Results

The solution here presented was implemented successfully and it's currently being used by thousands of users over the world. It has been proved that it is possible to take advantage of SAP HCM Processes & Forms to build an HR application, regardless of the front-end technology, and without jeopardizing performance. Additionally, bounded to this, there's the fact that new business logic added to SAP DITF will be automatically available on the application. This fact, along with fast-paced configuration required to setup new features in HCM P&F, proves high scalability and low-maintenance effort.

7.2 Critical Reception

The feedback obtained from application users were overall very good, with the new features standing out in comparison to the old application.



Data suggestion

Users were very pleased with the increase number of events that automatically fill the form, based on their input. This erased some of the confusion caused by the old application, which would require parallel conversations in order to decide what values should be entered/selected. Additionally, by not having to fill the form entirely, users saw their working time on the application reduced, specially requesters.



Data consistency

Since all the relevant SAP business rules are now leveraged by the application, there's no longer the issue of going through the full process just to find out later that the data is not accepted by SAP. Also, the persistence of data into SAP ERP is controlled by the SAP Administrator, which reduces the possibility of introducing incorrect data into the system.



Performance

As described above, this application fills in a big percentage of the form, according to user input. Obviously, having several events and a complex logic to suggest data may affect performance. So, the goal of this application is to find a balance between these

two things (suggest values vs. performance). This has been well-received by users, but requires continuous work.



Early error messages

At the beginning, some users were not satisfied with the amount of times error messages were displayed when they tried to submit a form. Remember that, in the old application, users would simply fill in the form and submit it with no worries about SAP business rules. Now, all the form is validated before it reaches SAP ERP. It was a customer decision to shift these rules, so users must comply with them.



Positive



Acceptable



Negative

7.3 Future work

This application was designed with the purpose of constantly evolving, by adding new form fields, new rules, etc. One of the points to be considered in the future is the migration of old infotype logic to the new infotype framework (DITF). Currently, only some infotypes have their logic in DITF, the remaining still use the old framework (module pools / user exits). In order to make these rules available for HCM P&F, these infotypes must be migrated.

Other point to take into consideration is the use of HCM P&F in other processes. The target of this product were employees, but processes can be built for OM objects as well. For instance, one that allows users to easily create new positions and assign them to the correct location in the company.

References

- Boes, S., 2007. *The SQL Trace (ST05) – Quick and Easy*. [Online]
Available at: <http://scn.sap.com/community/abap/testing-and-troubleshooting/blog/2007/09/05/the-sql-trace-st05-quick-and-easy>
[Accessed 12 February 2016].
- Dolinskaja, O., 2011. *Next Generation ABAP Runtime Analysis (SAT) – How to analyze performance*. [Online]
Available at: <http://scn.sap.com/community/abap/testing-and-troubleshooting/blog/2011/01/18/next-generation-abap-runtime-analysis-sat-how-to-analyze-performance>
[Accessed 12 February 2016].
- Garg, V. K. G. & Venkitakrishnan, N., 2003. *Enterprise Resource Planning - Concepts and Practice*. 2nd ed. s.l.:PHI Learning.
- Heuvelmans, W., Krouwels, A., Meijs, B. & Sommen, R., 2004. *Enhancing the Quality of ABAP Development*. 1 ed. s.l.:Galileo Press.
- konkconsulting, 2016. *konkconsulting*. [Online]
Available at: <http://www.konkconsulting.com>
[Accessed 4 January 2016].
- Krämer, C., Ringling, S. & Yang, S., 2006. In: *Mastering HR Management with SAP*. s.l.:Galileo Press.
- Marson, L., Mazhavanchery, M. & Murray, R., 2016. In: *SAP SuccessFactors Employee Central*. s.l.:Rheinwerk Publishing.
- Mistry, J., 2015. *The New SuccessFactors Employee Central Organisation Structure*. [Online]
Available at: <http://scn.sap.com/people/jaykumar.mistry/blog/2015/04/09/the-new-successfactors-employee-central-organisation-structure>
[Accessed 2 February 2016].
- Morgalis, J. & Toombs, B., 2013. In: *SAP ERP HCM Processes & Forms*. s.l.:Galileo Press.
- Oracle Corporation, 2014. *Oracle Self-Service Human Resources Data Sheet*. [Online]
Available at: <http://www.oracle.com/us/products/applications/057309.pdf>
[Accessed 5 February 2016].
- Oracle Corporation, 2015. *Oracle Self-Service Human Resources – Deploy Self-Service Capability Guide*. [Online]
Available at: http://docs.oracle.com/cd/V77972_01/current/acrobat/122sshrg.pdf
[Accessed 5 February 2016].
- Oracle Corporation, 2016a. *History | Oracle Corporation*. [Online]
Available at: <http://www.oracle.com/us/corporate/history/index.html>
[Accessed 4 February 2016].

Oracle Corporation, 2016b. *Oracle E-Business Suite | Applications*. [Online]
Available at: <http://www.oracle.com/us/products/applications/ebusiness/overview/index.html>
[Accessed 4 February 2016].

Osterwalder, A., 2004. *The Business Model Ontology: A proposition in a Design Science Approach*. [Online]
Available at: <http://www.uniempre.org.br/user-files/files/TheBusiness-Model-Ontology.pdf>
[Accessed 28 January 2016].

Panorama Consulting, 2015. *Clash of the Titans 2016 - An independent Comparison of SAP, Oracle and Microsoft Dynamics*. [Online]
Available at: <http://panorama-consulting.com/resource-center/clash-of-the-titans-2016-sap-vs-oracle-vs-microsoft-dynamics/>
[Accessed 19 February 2016].

Prabhu Mithal, R., 2010. *Performance Improvement in HCM P&F*. [Online]
Available at: <http://scn.sap.com/people/raghavendra.prabhu/blog/2010/10/21/performance-improvement-in-hcm-pf>
[Accessed 12 February 2016].

Rajora, S., 2012. *HCM Processes and Forms with new User Interface*. [Online]
Available at: <http://scn.sap.com/community/erp/hcm/employee-self-service/blog/2012/06/27/hcm-processes-and-forms-with-new-user-interface>
[Accessed 1 February 2016].

SAP SE or an SAP affiliate company, 2015. *Employee Central Master Implementation Guide (Document Version: Q4 2015)*. [Online]
Available at: <https://websmp210.sap-ag.de/~sapidb/012002523100008617742014E>
[Accessed 2 February 2016].

SAP SE, 2010a. *1043692 - Handling Time Constraint 3 Infotypes in HR Processes & Forms*. [Online]
Available at: <https://launchpad.support.sap.com/#/notes/0001043692>
[Accessed 10 October 2016].

SAP SE, 2011. *SAP to Accelerate Cloud Strategy with Acquisition of SuccessFactors*. [Online]
Available at: <http://global.sap.com/corporate-en/news.epx?pressid=17902>
[Accessed 2 February 2016].

SAP SE, 2016a. *SAP Enterprise Management & ERP Product*. [Online]
Available at: <http://go.sap.com/product/enterprise-management.html>
[Accessed 4 January 2016].

SAP SE, 2016b. *Personnel Administration (PA-PA)*. [Online]
Available at:
https://help.sap.com/saphelp_46c/helpdata/en/26/837202418e11d2959c00a0c9308b52/frameset.htm
[Accessed 7 January 2016].

SAP SE, 2016c. *Organizational Management in Personnel Administration*. [Online]
Available at:

https://help.sap.com/saphelp_46c/helpdata/en/48/35c4044abf11d18a0f0000e816ae6e/content.htm
[Accessed 16 January 2016].

SAP SE, 2016d. *Enterprise Structure*. [Online]

Available at:

https://help.sap.com/saphelp_46c/helpdata/en/48/35c4114abf11d18a0f0000e816ae6e/content.htm
[Accessed 16 January 2016].

SAP SE, 2016e. *Personnel Structure*. [Online]

Available at:

https://help.sap.com/saphelp_46c/helpdata/en/48/35c42b4abf11d18a0f0000e816ae6e/content.htm
[Accessed 16 January 2016].

SAP SE, 2016f. *Pay Scale Structure*. [Online]

Available at:

https://help.sap.com/saphelp_46c/helpdata/en/48/35c4524abf11d18a0f0000e816ae6e/content.htm
[Accessed 16 January 2016].

SAP SE, 2016g. *Elements of Pay Scale Structure*. [Online]

Available at:

https://help.sap.com/saphelp_46c/helpdata/en/48/35c45f4abf11d18a0f0000e816ae6e/content.htm?frameset=/en/48/35c4524abf11d18a0f0000e816ae6e/frameset.htm¤t_toc=/en/48/35d4ad4abf11d18a0f0000e816ae6e/plain.htm&node_id=18
[Accessed 16 January 2016].

SAP SE, 2016h. *Organizational Units*. [Online]

Available at:

http://help.sap.com/saphelp_46c/helpdata/en/bb/bdb041575911d189240000e8323d3a/content.htm
[Accessed 24 January 2016].

SAP SE, 2016i. *Job*. [Online]

Available at:

http://help.sap.com/saphelp_46c/helpdata/en/bb/bdb08f575911d189240000e8323d3a/content.htm?frameset=/en/bb/bdb08f575911d189240000e8323d3a/frameset.htm¤t_toc=/en/2c/283037301f327ae10000009b38f839/plain.htm&node_id=8
[Accessed 24 January 2016].

SAP SE, 2016j. *HCM Processes and Forms*. [Online]

Available at:

http://help.sap.com/saphelp_erp60_sp/helpdata/en/82/b6b94278560c31e10000000a1550b0/content.htm
[Accessed 26 January 2016].

SAP SE, 2016k. *HCM, Administrative Services 01*. [Online]

Available at:

http://help.sap.com/erp_mss_addon10/helpdata/en/37/74092f904e4311bcbc27696255d44f/content.htm
[Accessed 28 January 2016].

SAP SE, 2016l. *Process - HCM Processes and Forms*. [Online]

Available at:

http://help.sap.com/saphelp_erp60_sp/helpdata/en/4f/2a75713bcd18c3e1000000a421937/content.htm?frameset=/en/4f/423ed8132117a6e1000000a42189d/frameset.htm¤t_toc=/en/50/cedc53b5ef424de1000000a174cb4/plain.htm&node_id=7

[Accessed 1 February 2016].

SAP SE, 2016m. *Form Scenario - HCM Processes and Forms*. [Online]

Available at:

http://help.sap.com/saphelp_erp60_sp/helpdata/en/4f/2a74a33bcd18c3e1000000a421937/content.htm?frameset=/en/4f/423ed8132117a6e1000000a42189d/frameset.htm¤t_toc=/en/50/cedc53b5ef424de1000000a174cb4/plain.htm&node_id=16

[Accessed 1 February 2016].

SAP SE, 2016n. *Business Workflow for HCM Processes and Forms*. [Online]

Available at:

http://help.sap.com/saphelp_erp60_sp/helpdata/en/25/ccdc53b5ef424de1000000a174cb4/content.htm?frameset=/en/4f/9a39f5894d0cdf1000000a42189d/frameset.htm¤t_toc=/en/50/cedc53b5ef424de1000000a174cb4/plain.htm&node_id=17&show_children=false

[Accessed 1 February 2016].

SAP SE, 2016o. *HCM Processes and Forms (Technical Overview)*. [Online]

Available at:

https://help.sap.com/saphelp_erp60_sp/helpdata/en/42/f273461e5132c3e1000000a1553f6/content.htm

[Accessed 1 February 2016].

SAP SE, 2016p. *Configuration of Back-End Services*. [Online]

Available at:

https://help.sap.com/erp_hcm_ias_2013_01/helpdata/en/46/5afc46ebf05e40e1000000a11466f/content.htm?frameset=/en/46/5afd06ebf05e40e1000000a11466f/frameset.htm¤t_toc=/en/86/b6b94278560c31e1000000a1550b0/plain.htm&node_id=91&show_children=false

[Accessed 1 February 2016].

SAP SE, 2016r. *Process Object - HCM Processes & Forms*. [Online]

Available at:

https://help.sap.com/saphelp_erp60_sp/helpdata/en/ef/cbdc53b5ef424de1000000a174cb4/content.htm?frameset=/en/42/f273461e5132c3e1000000a1553f6/frameset.htm¤t_toc=/en/50/cedc53b5ef424de1000000a174cb4/plain.htm&node_id=8

[Accessed 9 February 2016].

SAP SE, 2016s. *Using Work Items*. [Online]

Available at:

http://help.sap.com/saphelp_erp60_sp/helpdata/en/8d/25f30f454311d189430000e829fbbd/content.htm

[Accessed 4 October 2016].

SAP SE, 2016t. *Enhancing the Components of Global Classes or Interfaces*. [Online]

Available at:

https://help.sap.com/saphelp_nw70ehp2/helpdata/en/86/b83142680d5c33e1000000a155106/content.htm

[Accessed 5 October 2016].

SAP SE, 2016u. *Code Inspector - ABAP Test and Analysis Tools*. [Online]

Available at:

https://help.sap.com/saphelp_nw74/helpdata/en/49/205531d0fc14cfe1000000a42189b/content.htm

[Accessed 8 October 2016].

SAP SE, 2016u. *Workload Overview*. [Online]

Available at:

https://help.sap.com/saphelp_nw70/helpdata/en/21/2c8f38c7215428e10000009b38f8cf/content.htm

[Accessed 17 October 2016].

SAP SE, 2016v. *Decoupling Infotypes*. [Online]

Available at:

http://help.sap.com/saphelp_erp60_sp/helpdata/en/43/a503b963161bbfe1000000a1553f7/content.htm

[Accessed 10 October 2016].

SAP SE, 2016w. *RFC - Components of SAP Communication Technology*. [Online]

Available at:

https://help.sap.com/saphelp_nw70/helpdata/en/6f/1bd5b6a85b11d6b28500508b5d5211/frameset.htm

[Accessed 10 October 2016].

SAP SE, 2016x. *Displaying System Logs (SM21)*. [Online]

Available at:

https://help.sap.com/saphelp_nw74/helpdata/en/b1/f4652c0f4d4e8fa04e165d161e386f/content.htm

[Accessed 17 October 2016].

Solomon, C., 2008. *HCM Processes & Forms: Gotchas, Bugs and Other Curiosities*. [Online]

Available at: <https://blogs.sap.com/2008/06/30/hcm-processes-forms-gotchas-bugs-and-other-curiosities/>

[Accessed 17 October 2016].

Solomon, C., 2009. *HCM Processes & Forms: Why HCM P&F often fails before it even has a chance*.

[Online]

Available at: <http://scn.sap.com/community/erp/hcm/blog/2009/07/08/hcm-processes-forms-why-hcm-pf-often-fails-before-it-even-has-a-chance>

[Accessed 1 February 2016].

SuccessFactors, an SAP Company, 2016. *Lars Dalgaard | SuccessFactors*. [Online]

Available at: http://www.successfactors.com/en_gb/about-successfactors/management-team/lars-dalgaard.html

[Accessed 2 February 2016].

8 Appendices

8.1 Appendix 1 – Business Model Canvas

<p><i>Key Partners</i></p> <ul style="list-style-type: none"> • SAP SE • Sales Partners 	<p><i>Key Activities</i></p> <ul style="list-style-type: none"> • Software Development • Support and Consultancy • Product Improvement 	<p><i>Value Proposition</i></p> <ul style="list-style-type: none"> • User-friendly HR Management at the distance of a click • Standardization of HR Processes • HR Costs Reduction • Performance • Customization 	<p><i>Customer Relationships</i></p> <ul style="list-style-type: none"> • Automation (where possible) • Support/Assistance post Go-live 	<p><i>Customer Segments</i></p> <ul style="list-style-type: none"> • Multinational Enterprises • Oil & Gas Industry • Public Sector
<p><i>Key Resources</i></p> <ul style="list-style-type: none"> • SAP HCM Consultants • Software Developers • SAP Software 			<p><i>Channels</i></p> <ul style="list-style-type: none"> • Online (website/newsletters) • Sales Partners • SAP SE 	
<p><i>Cost Structure</i></p> <ul style="list-style-type: none"> • Personnel • Hardware 			<p><i>Revenue Streams</i></p> <ul style="list-style-type: none"> • Product Sales • Remote Team / Outsourcing • Post Go-live Support 	

8.2 Appendix 2 – AHP

The first phase of AHP is to define the problem at hand. In this case, the goal is to select a valid application to manage HR processes.

The criteria used was:

- Performance (P)
- Configurability/Customization (C)
- Implementation Costs (saving) (I)
- Reliability (R)

To determine final criteria weights, the following matrix was used for comparison (P–C–I–R):

$$\begin{bmatrix} 1 & 3 & 2 & \frac{1}{2} \\ \frac{1}{3} & 1 & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 2 & 1 & \frac{1}{2} \\ \frac{1}{2} & 4 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 0,26 & 0,30 & 0,36 & 0,22 \\ 0,09 & 0,10 & 0,09 & 0,11 \\ 0,13 & 0,20 & 0,18 & 0,22 \\ 0,52 & 0,40 & 0,36 & 0,44 \end{bmatrix} = \begin{bmatrix} 0,29 \\ 0,10 \\ 0,18 \\ 0,43 \end{bmatrix}$$

Criteria Weights:

Performance	0,29
Configurability/Customization	0,10
Implementation Costs	0,18
Reliability	0,43

There are three alternatives being considered here:

- Enhanced HCM P&F (the solution described in this document) (H)
- Oracle SSHR (O)
- SuccessFactors EC (S)

The next step was to determine the weights of each alternative per criteria (i.e., rank alternatives by criteria).

Performance (H–O–S):

$$\begin{bmatrix} 1 & 1 & \frac{1}{2} \\ 1 & 1 & \frac{1}{2} \\ 2 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 0,25 & 0,25 & 0,25 \\ 0,25 & 0,25 & 0,25 \\ 0,50 & 0,50 & 0,50 \end{bmatrix} = \begin{bmatrix} 0,25 \\ 0,25 \\ 0,50 \end{bmatrix}$$

Configurability/Customization (H–O–S):

$$\begin{bmatrix} 1 & 4 & 5 \\ \frac{1}{4} & 1 & 2 \\ \frac{1}{5} & \frac{1}{2} & 1 \end{bmatrix} = \begin{bmatrix} 0,69 & 0,73 & 0,62 \\ 0,17 & 0,18 & 0,25 \\ 0,14 & 0,09 & 0,13 \end{bmatrix} = \begin{bmatrix} 0,68 \\ 0,20 \\ 0,12 \end{bmatrix}$$

Implementation Costs (H–O–S):

$$\begin{bmatrix} 1 & 2 & \frac{1}{5} \\ 1 & 1 & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{4} \\ 5 & 4 & 1 \end{bmatrix} = \begin{bmatrix} 0,22 & 0,29 & 0,21 \\ 0,11 & 0,14 & 0,16 \\ 0,67 & 0,57 & 0,63 \end{bmatrix} = \begin{bmatrix} 0,24 \\ 0,14 \\ 0,62 \end{bmatrix}$$

Reliability (H–O–S):

$$\begin{bmatrix} 1 & \frac{1}{2} & 4 \\ \frac{1}{2} & 1 & 5 \\ \frac{1}{4} & \frac{1}{5} & 1 \end{bmatrix} = \begin{bmatrix} 0,31 & 0,29 & 0,40 \\ 0,61 & 0,59 & 0,50 \\ 0,08 & 0,12 & 0,10 \end{bmatrix} = \begin{bmatrix} 0,33 \\ 0,57 \\ 0,10 \end{bmatrix}$$

In order to determine the absolute ranking of these alternatives, we must combine them with criteria weights, as below:

$$\begin{bmatrix} 0,25 & 0,68 & 0,24 & 0,33 \\ 0,25 & 0,20 & 0,14 & 0,57 \\ 0,50 & 0,12 & 0,62 & 0,10 \end{bmatrix} * \begin{bmatrix} 0,29 \\ 0,10 \\ 0,18 \\ 0,43 \end{bmatrix} = \begin{bmatrix} 0,33 \\ 0,36 \\ 0,31 \end{bmatrix}$$

In conclusion, here's the weight of each alternative:

Enhanced HCM P&F	0,33
Oracle SSHR	0,36
SuccessFactors EC	0,31

8.3 Appendix 3 – SAP HCM P&F Analysis

ISR_PROCESS_EVENT

ISR_PROCESS_EVENT is SAP's wrapper to all HCM P&F functionalities. The following table contains this function's signature. Highlighted are the most relevant parameters, mandatory if we want to replicate SAP calls.

Parameter Type	Parameter	Description
Import	SCENARIO	ISR Scenario (link to the Form/Process)
Import	MODE	Possible modes are: Create, Display and Modify
Import	NOTIF_NO	Notifications related (irrelevant)
Import	EVENT	Event Name (e.g. Check, Send)
Import	FLAG_INOUT_CONVERSION	Conversion of numeric values (Active/Inactive)
Import	FLAG_RESET	Refresh data in every call (Yes/No)
Import	ISR_PAGE_IN	Current Form Page (applicable if using several Forms)
Export	NOTIF_NO_OUT	Notifications related (irrelevant)
Export	RETURN	Error output
Export	ISR_FORM_VIEW	Form step's type (Request, Process, Approve)
Export	ISR_PAGE_OUT	New Form Page (applicable if using several Forms)
Changing	DATA	Form Data
Changing	ADDITIONAL_DATA	Form Possible Values
Changing	EXTERNAL_DATA	Form Header Values
Changing	MESSAGE_LIST	Messages (error, warning, success)
Changing	UI_ATTRIBUTES	Fields Security

Table 28 – ISR_PROCESS_EVENT Signature

Performance Analysis

In order to evaluate that the changes made in the HCM P&F proof of concept took effect, the following samples were used (these samples correspond to the execution time in seconds that the system took to create a new HR process/ticket):

Employee	Before	After
1	40,2	25,6
2	35,1	23,7
3	37,8	24,7
4	43,3	27,1
5	34,3	22,1
6	33,5	19,7
7	34,1	22,9
8	39,8	24,6
9	45,2	28,9
10	32,6	19,6
11	33,5	21,7
12	35,6	21,4

Table 29 – HCM P&F Solution Analysis – Samples

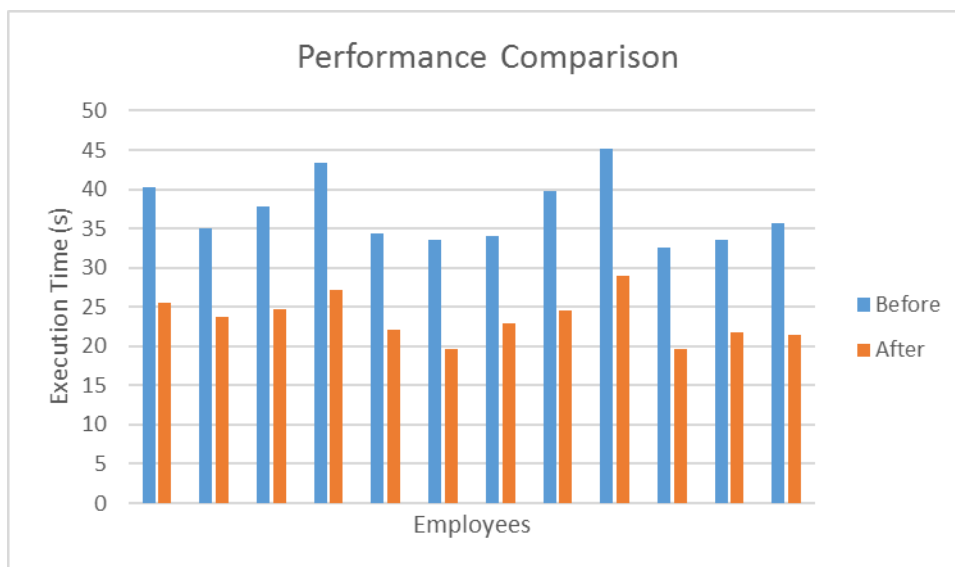


Figure 74 – HCM P&F Solution Analysis – Graphic

Using the paired-sample t test, these are the results:

SUMMARY		Alpha			0,05		Hyp Mean	0	
Groups	Count	Mean	Std Dev	Std Err	t	df	Cohen d	Effect r	
Before	12	37,08333	4,16366						
After	12	23,5	2,841894						
Difference	12	13,58333	1,768838	0,51062	26,60167	11	7,679241	0,992317	

T TEST

	p-value	t-crit	lower	upper	sig
One Tail	1,23E-11	1,795885			yes
Two Tail	2,46E-11	2,200985	12,45947	14,7072	yes

As displayed in column *sig*, since $p\text{-value} = 2,46E-11 < \alpha = 0,05$, it's possible to conclude with 95% confidence that the difference between the two samples is not due solely to chance.

In average, the execution time was reduced in 13,58 seconds, which corresponds to a 36% improvement.

Since the difference measures are reasonably symmetric, these results are considered valid:

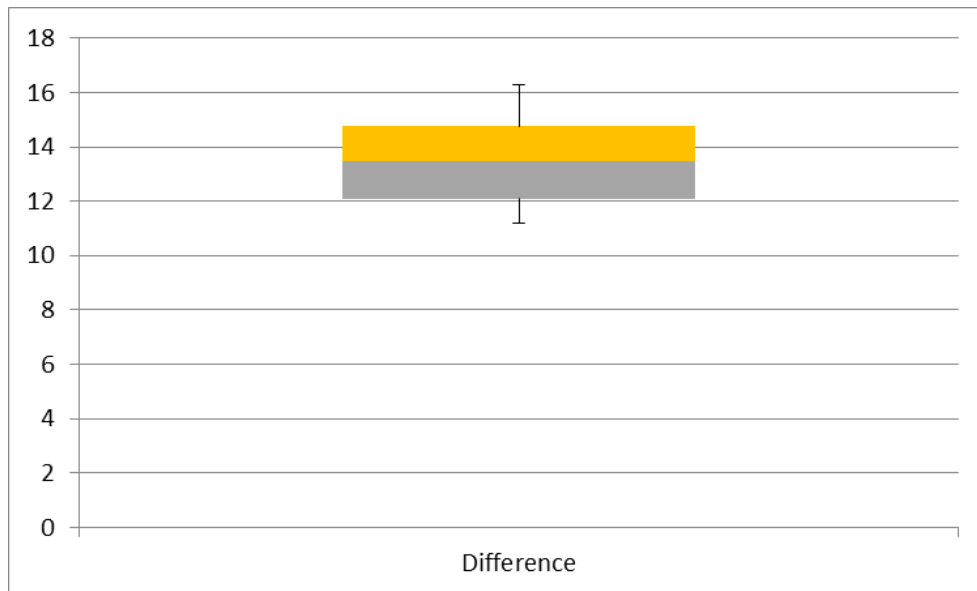


Figure 75 – HCM P&F Solution Analysis – Box Plot

8.4 Appendix 4 – Use Cases

Use Case 1 – Submit a New Ticket	
Actors	Requester
Description	This use case allows the actor to propose employee data changes by creating a ticket, filling in the necessary data and submitting it for approval.
Pre-conditions	1 – Actor is authenticated 2 – Actor has authorizations to create a new ticket
Post Conditions	Ticket is created, form data is saved and it is now pending for subsequent actors

Table 30 – Use Case 1 – Submit a New Ticket

Actor Actions	System Actions
1 – Select ticket type, employee, country of assignment and create ticket 2 – Edit form data 3 – Submit	1 – Start process in SAP 2 – Display employee current data 3 – Fetch default values/possible values 4 – Execute business validations 5 – Save form data

Table 31 – Event Flow – Submit a New Ticket

Use Case 2 – Manage Compensation Data	
Actors	Compensation/Benefits Manager
Description	This use case allows the actor to validate data on the form and add salary-related information.
Pre-conditions	1 – Actor is authenticated 2 – Ticket is pending for the actor in <i>Edit Form</i> status
Post Conditions	Form data is saved and ticket is now pending approval from Managers

Table 32 – Use Case 2 – Manage Compensation Data

Actor Actions	System Actions
1 – Open Ticket 2 – Edit Form 3 – Submit	1 – Execute business validations 2 – Save form data 3 – Ticket status is updated

Table 33 – Event Flow – Manage Compensation Data

Use Case 3 – Approve Ticket	
Actors	HR Manager Direct Manager
Description	This use case allows actors to approve a ticket
Pre-conditions	1 – Actor(s) is(are) authenticated 2 – Ticket is pending for the actor(s) in <i>Approve Form</i> status
Post Conditions	Ticket is now pending for the subsequent actor

Table 34 – Use Case 3 – Approve Ticket

Actor Actions	System Actions
1 – Open Ticket 2 – Check form data 3 – Approve	1 – Ticket status is updated

Table 35 – Event Flow – Approve Ticket

Use Case 4 – Reject Ticket	
Actors	HR Manager Direct Manager SAP Administrator
Description	This use case allows actors to reject a ticket
Pre-conditions	1 – Actor(s) is(are) authenticated 2 – Ticket is pending for the actor(s)
Post Conditions	Ticket is now pending for the Requester, waiting for corrections

Table 36 – Use Case 4 – Reject Ticket

Actor Actions	System Actions
1 – Open Ticket 2 – Check form data 3 – Approve	1 – Ticket status is updated

Table 37 – Event Flow – Reject Ticket

Use Case 5 – Post to SAP	
Actors	SAP Administrator
Description	This use case allows the actor to post the form data into SAP, using a semi-automated process
Pre-conditions	1 – Actor is authenticated 2 – Ticket is pending for the actor in <i>SAP Assisted Posting</i> status
Post Conditions	Employee record in SAP is updated with form data and ticket is now in status <i>Review Form</i>

Table 38 – Use Case 5 – Post to SAP

Actor Actions	System Actions
1 – Open Ticket 2 – Execute semi-automated actions within SAP 3 – Manually correct data if necessary 4 – Close Assisted Posting	1 – Execute business validations 2 – Save employee data 3 – Ticket status is updated

Table 39 – Event Flow – Post to SAP

Use Case 6 – Close Ticket	
Actors	SAP Administrator Payroll Administrator
Description	This use case allows the actors to close the ticket, acknowledging that changes were persisted into SAP
Pre-conditions	1 – Actor is authenticated 2 – Ticket is in status <i>Review Form</i> and pending for the Actor
Post Conditions	Ticket status is now Closed

Table 40 – Use Case 6 – Close Ticket

Actor Actions	System Actions
1 – Open Ticket 2 – Acknowledge	1 – Ticket status is updated

Table 41 – Event Flow – Close Ticket