

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки,  
Факультет комп'ютерних наук і технологій  
(повне найменування інституту, факультету)

Кафедра програмних засобів  
(повне найменування кафедри)

## Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРАКТИЧНЕ ВИКОРИСТАННЯ  
ХМАРНОЇ ПЛАТФОРМИ SAP ДЛЯ РОЗРОБКИ  
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ  
RESEARCH AND PRACTICAL APPLICATION OF SAP  
CLOUD PLATFORM FOR SOFTWARE DEVELOPMENT

Виконав: студент(ка) 2 курсу, групи КНТ-219м  
Спеціальності \_\_\_\_\_

122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Системи штучного інтелекту

Полякова К.С.

(прізвище та ініціали)

Керівник Пархоменко А.В.

(прізвище та ініціали)

Рецензент Морщавка С.В.

(прізвище та ініціали)



## 6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-3 Основна частина	Пархоменко А.В., доцент		
4 Економіко-організаційна частина	Пожуєва Т.О., професор		
5 Охорона праці та безпека в надзвичайних ситуаціях	Коробко О.В., ст. викладач		
Нормоконтролер	Камінська Ж.К., асистент		

7. Дата видачі завдання “04” вересня 2020 року.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2–3 тижні	Розділ 1
3	Дослідження та удосконалення методів, моделей й алгоритмів вирішення задачі.	4–5 тижні	Розділ 2
4	Розробка архітектури системи.	6 тиждень	Розділ 3
5	Розробка програмного забезпечення.	7–8 тижні	Розділ 3
6	Тестування та експериментальне дослідження програмного забезпечення.	9 тиждень	Розділ 3
7	Розробка економіко-організаційної частини.	10 тиждень	Розділ 4
8	Розробка заходів з охорони праці та безпеки в надзвичайних ситуаціях.	11 тиждень	Розділ 5
9	Оформлення пояснювальної записки та документів до неї.	12-13 тиждень	Додатки
10	Нормоконтроль та рецензування.	14–15 тижні	
11	Захист роботи.	16 тиждень	

Студент(ка)

\_\_\_\_\_  
(підпис) Полякова К.С.  
(прізвище та ініціали)

Керівник проєкту (роботи)

\_\_\_\_\_  
(підпис) Пархоменко А.В.  
(прізвище та ініціали)

## РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:  
145 с., 20 рис., 9 табл., 3 дод., 40 джерел.

ІНТЕЛЕКТУАЛЬНЕ ПІДПРИЄМСТВО, ІТ-ЛАНДШАФТ, ХМАРНИЙ  
СЕРВІС, МЕТОДОЛОГІЯ РОЗРОБКИ, БІЗНЕС-ЗАСТОСУНОК,  
ДОКУМЕНТООБІГ, НАКАЗ НА ПЕРЕВЕДЕННЯ.

Об'єкт дослідження – автоматизація документообігу підприємства.

Предмет дослідження – хмарні сервіси для розробки та експлуатації програмного забезпечення для адміністрування персоналу підприємства.

Мета роботи – дослідження та практичне використання хмарних сервісів для підвищення рівня автоматизації документообігу інтелектуального підприємства.

Матеріали, методи та технічні засоби: SAP Cloud Platform, методологія DevOps, фреймворк CALMS, модель BYOL, мова програмування АВАР 4.

Результати. З використанням SAP Cloud Platform та мови програмування АВАР/4 розроблено бізнес-застосунок для автоматичного формування документів (наказ про переведення окремого працівника або групи працівників), що пов'язані з іншими складовими системи документообігу підприємства (табелі, штатний розклад, тощо).

Висновки. Розроблено програмний продукт, що дозволяє підвищити рівень автоматизації документообігу підприємства за рахунок зменшення витрат часу на формування наказів на переведення персоналу та внесення змін у суміжні документи.

Галузь використання – корпоративна система управління підприємством SAP ERP.

## ABSTRACT

Explanatory note to the master thesis: 145 pages, 20 figures, 9 tables, 3 appendixes, 40 sources.

INTELLECTUAL ENTERPRISE, IT LANDSCAPE, CLOUD SERVICE, DEVELOPMENT METHODOLOGY, BUSINESS APPLICATION, DOCUMENT FLOW, ORDER FOR TRANSFER.

The object of study is the automation of enterprise document flow.

The subject of the study are cloud services for the development and operation of software for the administration of enterprise personnel.

The goal of the master thesis is research and practical usage of cloud services to increase the level automation of the document flow of an intellectual enterprise.

Materials, methods, and tools: SAP Cloud Platform, DevOps methodology, CALMS framework, BYOL model, AVAR 4 programming language.

Results. The business application for automatic documents generation (an order to transfer an individual employee or group of employees), which are related to other components of the document flow system of the enterprise (report card, staffing, etc.) is developed using the SAP Cloud Platform and the AVAR / 4 programming language.

Conclusions. A software product has been developed that allows to increase the level of automation of the document flow of the enterprise by reducing the time spent on the formation of orders for the transfer of personnel and changes in related documents.

The field of application is corporate enterprise resource planning system SAP ERP.

## ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Дослідження існуючих моделей надання хмарних сервісів.....	10
1.1 Інфраструктура як послуга.....	10
1.2 Платформа як послуга.....	12
1.3 Програмне забезпечення як послуга.....	13
1.4 Постановка задач дипломної кваліфікаційної роботи магістра	14
2 Аналіз хмарних рішень для розробки та експлуатації програмного забезпечення.....	15
2.1 Огляд цифрових сервісів SAP.....	15
2.2 Дослідження архітектури та функціональних можливостей SAP Cloud platform.....	16
2.3 Методологія DevOps.....	20
2.4 Огляд хмарних рішень на основі SAP HANA.....	27
2.5 Висновки до розділу 2.....	32
3 Розробка бізнес-застосунку на SAP Cloud Platform.....	33
3.1 Аналіз технічного завдання.....	33
3.2 Вибір інструментарію розробки.....	45
3.3 Методика розробки хмарного застосунку.....	47
3.4 Особливості програмної реалізації.....	55
3.5 Висновки до розділу 3.....	59
4. Економіко-організаційна частина.....	60
4.1 Планування розробки програмного продукту.....	60
4.2 Визначення витрат на розробку програми.....	62
4.2.1 Розрахунок основної заробітної плати.....	63
4.2.2 Розрахунок додаткової заробітної плати.....	63
4.2.3 Відрахування на єдиний соціальний внесок.....	64

4.2.4	Визначення витрат на матеріали.....	64
4.2.5	Витрати на спеціальне обладнання.....	65
4.2.6	Розрахунок накладних витрат.....	68
4.3	Ціна ПЗ.....	69
4.4	Розрахунок економічної ефективності програмного продукту...	71
4.5	Висновки до розділу 4.....	73
5.	Охорона праці та безпека в надзвичайних ситуаціях.....	74
5.1	Аналіз потенційних небезпек.....	74
5.2	Заходи із забезпечення безпеки.....	75
5.3	Заходи з виробничої санітарії та гігієни праці.....	79
5.4	Заходи безпеки у надзвичайних ситуаціях.....	86
5.4.1	Заходи з пожежної безпеки.....	86
5.4.2	Заходи з цивільного захисту.....	87
5.5	Висновки до розділу 5.....	89
	Висновки.....	90
	Перелік джерел посилання.....	91
	Додаток А Текст програми.....	96
	Додаток Б Результати роботи програми.....	133
	Додаток В Слайди презентації.....	138

**ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК**

API	– Application Programming Interface;
BYOL	– Bring Your Own Language;
CALMS	– Culture, Automation, Lean, Measurement, Sharing;
DevOps	– Development Operation;
ERP	– Enterprise Resource Planning;
IaaS	– Infrastructure-as-a-Service;
IoT	– Internet of Things;
PaaS	– Platform-as-a-Service;
SaaS	– Software-as-a-Service;
SAP HEC	– SAP HANA Enterprise Cloud;
SCP	– SAP Cloud Platform;
SHCP	– SAP HANA Cloud Platform
ЕМІ	– електромагнітний імпульс;
ІТ	– інформаційні технології;
НС	– надзвичайні ситуації;
ОЦ	– обчислювальний центр;
ПЗ	– програмне забезпечення;
ПК	– персональний комп'ютер.
ПУЕ	– правила улаштування електроустановок;
ЕОМ	– електронно-обчислювальна машина.

## ВСТУП

Існуюча практика фрагментації обробки даних в рамках різних процесів, систем, застосунків, видів діяльності та груп користувачів викликає ефект комбінаційного вибуху, в якому кожен нові отримані дані збільшують число взаємозв'язків в геометричній прогресії. Підвищення різноманіття складу обчислювальної інфраструктури викликає надмірну складність її операційної експлуатації і як наслідок призводить до повної паралізації управління і обробки інформації. В той же час хмарні технології дозволяють вийти за рамки ізольованих систем, ефективно долаючи інформаційний бар'єр і підвищуючи технологічний рівень інтелектуального підприємства.

Управління великими масивами даних, роботизація рутинних операцій, аналітика, машинне навчання – це сучасні технології, які можуть використовувати підприємства та компанії для реалізації інноваційних сценаріїв в бізнесі. Однак, їх впровадження вимагає значних зусиль, крім того, вони погано інтегруються, а інформація про них розкидана в безлічі джерел в Інтернеті і часто не є актуальною. Щоб відповідально побудувати сучасне цифрове рішення масштабу підприємства, системні адміністратори та архітектори повинні володіти безліччю умінь і величезним практичним досвідом. Вирішити таку задачу допомагають готові хмарні платформи, що зібрані великими постачальниками програмного забезпечення (ПЗ) і сервісів та надаються за підпискою [1].

Метою роботи є дослідження та практичне використання хмарних сервісів для підвищення рівня автоматизації документообігу інтелектуального підприємства.

# 1 ДОСЛІДЖЕННЯ ІСНУЮЧИХ МОДЕЛЕЙ НАДАННЯ ХМАРНИХ СЕРВІСІВ

Інфраструктура як послуга (Infrastructure-as-a-Service - IaaS), Платформа як послуга (Platform-as-a-Service - PaaS) та Програмне забезпечення як послуга (Software-as-a-Service - SaaS) - це моделі надання хмарних сервісів. Їх співвідношення можна представити у вигляді піраміди з різним рівнем контролю інформації. Вершина - це кінцевий користувач, який працює з особистими даними, представленими у вигляді програми або сервісу з зручним інтерфейсом. Програма або сервіс розгортаються на якійсь технологічній платформі - це другий рівень піраміди. Нарешті, її основа - це інфраструктура: віртуальні сервери, обчислювальні потужності, накопичувачі і канали зв'язку [2].



Рисунок 1.1 – Піраміда хмарних сервісів [2]

## 1.1 Інфраструктура як послуга

Інфраструктура як послуга (IaaS) – це хмарне рішення, в рамках якого постачальник надає користувачам доступ до обчислювальних ресурсів, таких як сервери, сховище та мережі з оплатою за фактом використання.

Організації використовують свої власні платформи та застосунки в інфраструктурі постачальника послуг. Основні функції та особливості платформи наступні:

- замість того щоб купувати обладнання і негайно оплачувати його, користувачі оплачують IaaS за запитом;
- інфраструктура масштабується залежно від потреб в обчислювальній потужності і пам'яті;
- підприємства економлять на витратах на покупку і обслуговування свого власного апаратного забезпечення;
- дані знаходяться в хмарі, тому немає єдиної точки відмови;
- підтримується віртуалізація завдань адміністрування, що дозволяє вивільнити час для виконання іншої роботи [3].

При цій моделі споживач отримує інформаційно-технологічні ресурси – віртуальні сервери з певною обчислювальною потужністю та обсягами пам'яті. Всім «залізом» займається провайдер. Він встановлює на нього ПО для створення віртуальних машин, але не займається установкою і підтримкою ПЗ користувача. Провайдер контролює тільки фізичну та віртуальну інфраструктуру. Приклади IaaS: IBM Softlayer, Hetzner Cloud, Microsoft Azure, Amazon EC2, GigaCloud. Клієнти IaaS – це системні адміністратори компаній [2].

Є декілька варіантів використання, в яких може бути вигідна ця модель послуг. Компанії, які не мають власного центру обробки даних, розглядають IaaS як швидко і дешево інфраструктуру для своїх бізнес-проектів, які при необхідності можна розширити або припинити. Інший приклад використання IaaS - це традиційні компанії, яким потрібні обчислювальні потужності для обробки змінних робочих навантажень з меншими капітальними витратами. В обох випадках компанії будуть платити тільки за послуги, які вони отримують [3].

## 1.2 Платформа як послуга

Платформа як послуга (PaaS) - це хмарне рішення, в рамках якого користувачам надається хмарне середовище, в якому вони можуть здійснювати розробку та експлуатацію застосунків. Крім сховища та інших обчислювальних ресурсів, користувачі можуть використовувати готові інструменти для розробки, налаштування та тестування своїх власних застосунків. Підтримка базової інфраструктури здійснюється постачальником. Основні функції та особливості платформи наступні:

- PaaS надає платформу з інструментами для тестування, розробки і розміщення застосунків в тому ж середовищі;
- організації отримують можливість зосередитися на розробці, не турбуючись про базову інфраструктуру;
- постачальники керують захистом, операційними системами, серверним програмним забезпеченням і резервним копіюванням;
- полегшується спільна робота, навіть якщо співробітники працюють віддалено [3].

Таким чином, хмарний провайдер надає доступ до операційних систем, засобів розробки і тестування, систем управління базами даних. Провайдер контролює не тільки сервери, системи зберігання даних і обчислювальні потужності, але також пропонує користувачеві на вибір певні платформи і засоби управління ними. Приклади PaaS: Google App Engine, IBM Bluemix, Microsoft Azure, VMWare Cloud Foundry. Користувачі PaaS-сервісів - це розробники програмного забезпечення (ПЗ) [2].

Для компаній-розробників, які хотіли б реалізувати гнучкі методології розробки, найбільше підходить модель PaaS. Постачальники PaaS надають безліч послуг, які можна використовувати всередині застосунків. Ці послуги завжди будуть доступними і сучасними. PaaS забезпечує дуже простий спосіб тестування і створення прототипів нових застосунків. Це дозволяє економити гроші при розробці нових послуг і застосунків. Застосунки можна

випускати швидше, ніж зазвичай, для отримання відгуків користувачів. Application Programming Interface (API) – це нова парадигма в сфері розробки, і хмара надає досконалу платформу для її реалізації [3].

### **1.3 Програмне забезпечення як послуга**

Програмне забезпечення як послуга (SaaS) – це хмарне рішення, в рамках якого забезпечується доступ користувачів до хмарного програмного забезпечення постачальника. Його користувачі не встановлюють застосунки на свої власні локальні пристрої. Вони знаходяться у віддаленій хмарній мережі, доступ до якої здійснюється через вебінтерфейс або API. Застосунок дозволяє користувачам зберігати і аналізувати дані і спільно працювати над проектами. Основні функції та особливості платформи наступні:

- постачальники SaaS надають користувачам ПЗ та застосунки на основі передплати; користувачам не потрібно встановлювати або оновлювати ПЗ, а також керувати ним, оскільки це роблять постачальники;
- дані в хмарі захищені, тому збій обладнання не призводить до втрати даних;
- використання ресурсів масштабується залежно від потреб у послугах;
- програма є доступною майже з будь-якого пристрою, підключеного до мережі Інтернет і практично в будь-якій точці світу [3].

Програми та сервіси розробляє і обслуговує провайдер, розміщує їх в хмарі і пропонує кінцевому користувачеві через браузер або застосунок на його персональному комп'ютері (ПК). Клієнт лише вносить абонплату (або користується сервісом безкоштовно), а оновленням і технічною підтримкою програм займається провайдер. SaaS-сервіси можуть надавати місце для зберігання файлів (Dropbox), офісний пакет документів для роботи (Google Doc, Microsoft Office 365), допомагати організувати фотографії (Flickr) або спілкуватися з іншими людьми (Facebook) [2]. Сьогодні моделі надання

послуг SaaS є загальноприйнятими в багатьох компаніях, які хотіли б отримувати вигоду від використання застосунків без необхідності обслуговування та оновлення інфраструктури та компонентів. Застосунки для пошти, управління ресурсами підприємства (ERP – Enterprise Resource Planning), спільної роботи і офісні застосунки - найбільш затребувані рішення SaaS. Ця хмарна модель є найпоширенішою. Основні переваги моделі - це гнучкість і еластичність [3].

#### **1.4 Постановка задач дипломної кваліфікаційної роботи магістра**

Як показали проведені дослідження, з точки зору кінцевого користувача SaaS – найбільш зрозуміла та зручна модель надання хмарного сервісу. Часто простіше і ефективніше використовувати готовий SaaS-сервіс, який вже відповідає певним вимогам. Але готові рішення не завжди існують, і в такому випадку використовуються моделі PaaS і IaaS. Не існує універсального підходу до впровадження хмарних технологій для IT-ландшафту підприємства. Вибір моделі та процес переходу в хмару є нетривіальними задачами. Тому, тема магістерської роботи є актуальною.

Мета роботи – дослідження та практичне використання хмарних сервісів для підвищення рівня автоматизації документообігу інтелектуального підприємства.

Задачі роботи:

- дослідити особливості хмарних рішень компанії SAP;
- вивчити методологію DevOps та фреймворк CALMS для підвищення ефективності процесів розробки та експлуатації ПЗ;
- удосконалити методику розробки ПЗ на основі хмарного сервісу з використанням моделі BYOL для мови програмування ABAP 4;
- розробити та впровадити в експлуатацію бізнес - застосунок для автоматичного формування наказів по підприємству для адміністрування персоналу.

## **2 АНАЛІЗ ХМАРНИХ РІШЕНЬ ДЛЯ РОЗРОБКИ ТА ЕКСПЛУАТАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

### **2.1 Огляд цифрових сервісів SAP**

Компанія SAP - один з світових лідерів на ринку корпоративних застосунків. Компанія допомагає організаціям будь-якого розміру і спеціалізації ефективніше управляти своїм бізнесом. Рішеннями і сервісами SAP користується понад 345 000 клієнтів, передові технології компанії гарантують високу рентабельність, сприяють безперервній адаптації та стійкого зростання. Вона є також одним з відомих постачальників комплексних рішень для банків, підприємств, компаній в Україні.

Компанія пропонує цілу лінійку рішень для Інтернету речей, які входять в цифрову інноваційну систему SAP Leonardo. Зокрема, платформа SAP Leonardo об'єднує різні програмні можливості: машинне навчання, Інтернет речей, Big Data, аналітику і блокчейн на SAP Cloud Platform разом з досвідом SAP, глибоким знанням індустрії і просунутою методологією design thinking [4].

За допомогою SAP Cloud Platform підтримуються функції по роботі з великими даними і аналітикою, можливість пов'язувати людей, об'єкти і бізнес, а також такі технології, як машинне навчання, щоб дати можливість реалізувати стратегії для Інтернету речей і Індустрії 4.0 в сфері цифрової логістики, виробництві і управління активами [4].

SAP IoT Bridge - це цифровий командний центр, який дозволить операційним менеджерам забезпечити прозорість бізнес-процесів і дасть можливість діяти в режимі реального часу.

SAP Global Track and Trace - хмарний застосунок для уніфікованого трекінгу, моніторингу та отримання звітів про об'єкти та бізнес-процеси в мережевих каналах поставок.

SAP Digital Manufacturing Insights - хмарне рішення для централізованого управління виробництвом та оптимізації робочих процесів. SAP Asset Manager - хмарний мобільний застосунок, що дозволяє відслідковувати стан здоров'я співробітників, обладнання, управляти техобслуговуванням і системою безпеки. SAP Leonardo IoT Edge - хмарне програмне забезпечення для обчислень, зберігання даних і бізнес-семантики, за допомогою якого пристрої без доступу до дата-центру компанії можуть продовжувати ефективно брати участь у бізнес-процесах практично в режимі реального часу [4].

## **2.2 Дослідження архітектури та функціональних можливостей SAP Cloud platform**

Як показали проведені дослідження, PaaS відноситься до хмарних платформ, що забезпечують середовища виконання для розробки, тестування та управління застосунками. Постачальники PaaS надають повну інфраструктуру для розробки застосунків, тоді як розробники відповідають за код. Однією з таких платформ для швидкого та ефективного вирішення різних задач є SAP Cloud Platform (SCP).

Хмарна платформа SCP надає повний спектр сервісних каталогів, включаючи базу даних, сховище й резервне копіювання даних, сервіс звітів і рівень транзакцій для багатоплатформної розробки ПЗ. Клієнти платформи SCP можуть використовувати це хмарне середовище для управління розробкою ПЗ, або також можуть використовувати гібридну модель, засновану на хмарному середовищі і на власному середовищі розробки [5].

Платформа SCP може бути інтегрована з застосунками SAP, сторонніми застосунками або внутрішніми рішеннями компанії для отримання даних і програмної розробки (рис.2.1) [5].

SCP – це корпоративна хмарна платформа, за допомогою якої компанії можуть розробляти власні програми, розширення або доповнення

для вже існуючих продуктів SAP або ж створювати гібридний ІТ-ландшафт в компанії за допомогою інтеграції хмарних рішень.

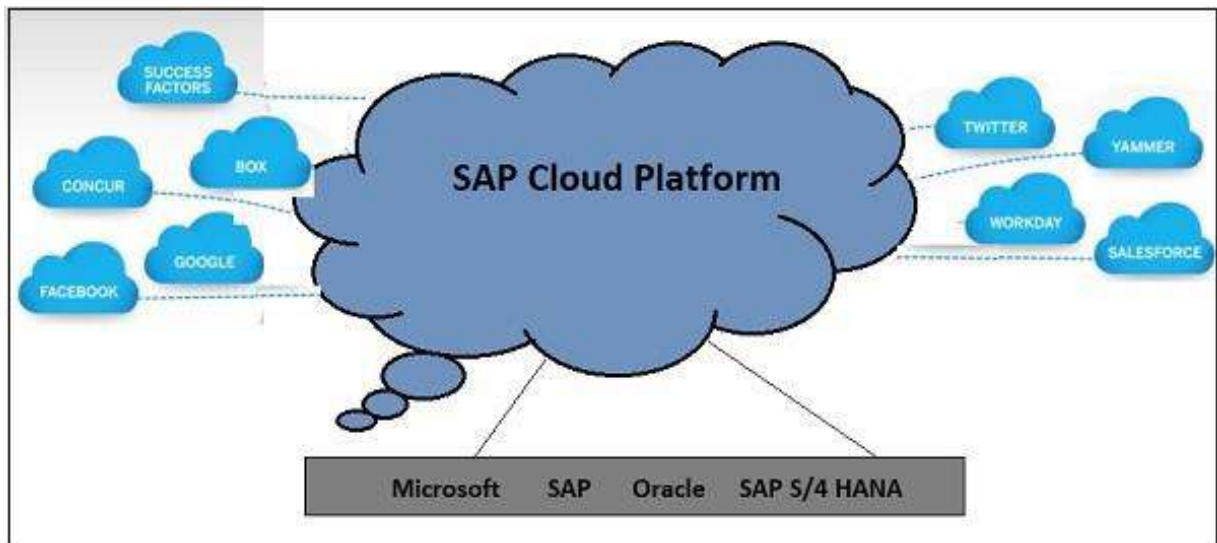


Рисунок 2.1 – Інтеграція SCP з іншими застосунками [5]

Хмарні технології від SAP відкривають безмежні можливості для розвитку ресурсів компанії, інструментів документообігу та мобільних хмарних сервісів. Також вони дають зелене світло використанню машинного навчання, великих даних, Інтернету речей та інших інноваційних технологій.

Платформа дозволяє створювати, об'єднувати й розширювати власні хмарні застосунки без часових і фінансових витрат на обслуговування ІТ-інфраструктури [6]. Можливості SCP:

- створення корпоративних мобільних застосунків;
- розробка персоналізованих мобільних і HTML-застосунків, які працюють в хмарі; доступ до них співробітники отримують як з робочого столу, так і зі смартфона;
- готові застосунки: портфель бізнес-застосунків SAP, а також партнерські сервіси та програми, побудовані на платформі SAP HANA, які охоплюють все - від побудови відносин з клієнтами до планування ресурсів підприємств;

- сервіси для спільної роботи з віддаленим доступом в масштабах всієї організації незалежно від її розміру;
- безшовна інтеграція всіх застосунків в єдину екосистему в хмарному або гібридному середовищі; застосування Application Programming Interface (API) як інструменту для взаємодії з клієнтами і партнерами, а також для виходу на нові ринки;
- аналітика бізнес-даних, аналіз інформації в режимі реального часу з використанням портфелю аналітичних рішень;
- розробка, розгортання та управління застосунками для Internet of Things (IoT), межмашинної взаємодії та дистанційної синхронізації даних в режимі реального часу;
- безпечне зберігання, масштабування, поширення та управління даними, з використанням всіх можливостей SCP;
- екосистема SAP - це відкрита торгова площа для нових застосунків SAP Customer Experience, незалежних постачальників ПЗ і інших виробників;
- вебсередовище для розробки WebIDE і інструмент для швидкого прототипування Build; підтримка розробки застосунків на Java, C ++, Python, XSJS, PHP, Ruby, Node. js, Go, .Net [6]. Послуги SCP:
  - цифрова трансформація, завдяки швидкому розгортанню застосунків наступного покоління на платформі SCP, яка є основою для SAP Leonardo;
  - відкрита PaaS хмара підприємства: єдина відкрита корпоративна PaaS платформа для задоволення всіх потреб підприємства;
  - управління даними та аналітика: використовує служби управління даними і аналітики для створення інтелектуального підприємства, що надають інформацію та дані в реальному часі для підтримки прийняття рішень або аналізу;

- інтеграція і координація: надійно інтегрує і безперешкодно координує бізнес-процеси, рішення й дані між хмарними і локальними застосунками в режимі реального часу;
- цифровий досвід: створює користувальницький досвід через цифрові точки дотику й оптимізує дизайн користувальницького досвіду і розробку, спільну роботу, взаємодію з користувачем і мобільність;
- розширюваність рішення: використовує розширюваність платформи, що дозволяє впроваджувати гнучкі рішення й інновації для бізнес-застосунків інтелектуального підприємства [7].

Для хостингу SAP Cloud Platform використовуються сертифіковані data-центри рівня Tier-3 і Tier-4 з глобальної підтримкою 24/7 і доступністю 99,9%. Data-центри SAP розташовані по всьому світу.

Деякі компанії розміщують цифрові рішення SAP в своїй фізичній інфраструктурі, оскільки бояться, що їх дані в хмарі недостатньо захищені. Однак віддалені сховища SAP захищені краще, ніж локальна інфраструктура, і дешевші у використанні. SAP гарантує повний захист даних «в хмарі», використовуючи кращі практики забезпечення безпеки в сфері ІТ, точну відповідність законодавству країни, в якій працює, і кращу інфраструктуру. Data-центри вищого рівня надійності Tier-3 і Tier-4 не припиняють свою роботу навіть у разі відмов і помилок. Крім того, працюють подвійне резервування і дублювання системи [6].

Хмарна платформа SAP заснована на відкритих стандартах і надає розробникам гнучкість і контроль над розгортанням хмар, застосунків і середовищ. SCP використовує різні середовища розробки, включаючи Neo і Cloud Foundry, і надає різні мови програмування.

Основним обмеженням хмарної платформи SAP є те, що вона призначена для підтримки інтеграції між хмарою і хмарою, але не для інтеграції між локальними системами. Це робить інструменти менш корисними для інтеграції застосунків SAP і сторонніх застосунків, які знаходяться як локальних серверах [7].

Таким чином, два головні завдання, які вирішує платформа SCP - це розробка та експлуатація IT-рішень (хостинг, моніторинг, масштабування і багато іншого) [8].

Архітектуру системи представлено на рис. 2.2 [8]. Серед сервісів, які є в SAP Cloud Platform, можна виділити кілька категорій:

- сервіси для обробки і зберігання даних: in-memory, SQL, NoSQL, Big Data Services;
- сервіси для розробки застосунків: Java, XS / JS (внутрішня мова SAP для програмування під SAP HANA), а також підтримка так званої моделі BYOL (Bring Your Own Language) - використання будь-якої мови програмування, наприклад, PHP, Python, ABAP 4;
- набір готових технологічних сервісів для створення бізнес-застосунків: блокчейн-сервіси, інтеграційна шина, Інтернет речей та інші
- набір готових бізнес-сервісів: торговий кошик, програма лояльності та інші.

За допомогою готових технологічних сервісів компанії можуть швидко розробляти бізнес-застосунки або ж купувати готові рішення у інших компаній для подальшої інтеграції [8].

Узагальнену карту сервісів SCP представлено на рис. 2.3.

Таким чином, незважаючи на те, що розроблені програми працюють на SCP з різними функціями та перевагами, у них є спільні характеристики, що дозволяє здійснювати цифрову трансформацію бізнесу.

## 2.3 Методологія DevOps

DevOps (DEVELOPMENT OPERATION) - це набір практик для підвищення ефективності процесів розробки (Development) і експлуатації (Operation) ПЗ за рахунок їх безперервної інтеграції та активної взаємодії профільних фахівців за допомогою інструментів автоматизації (рис. 2.4).

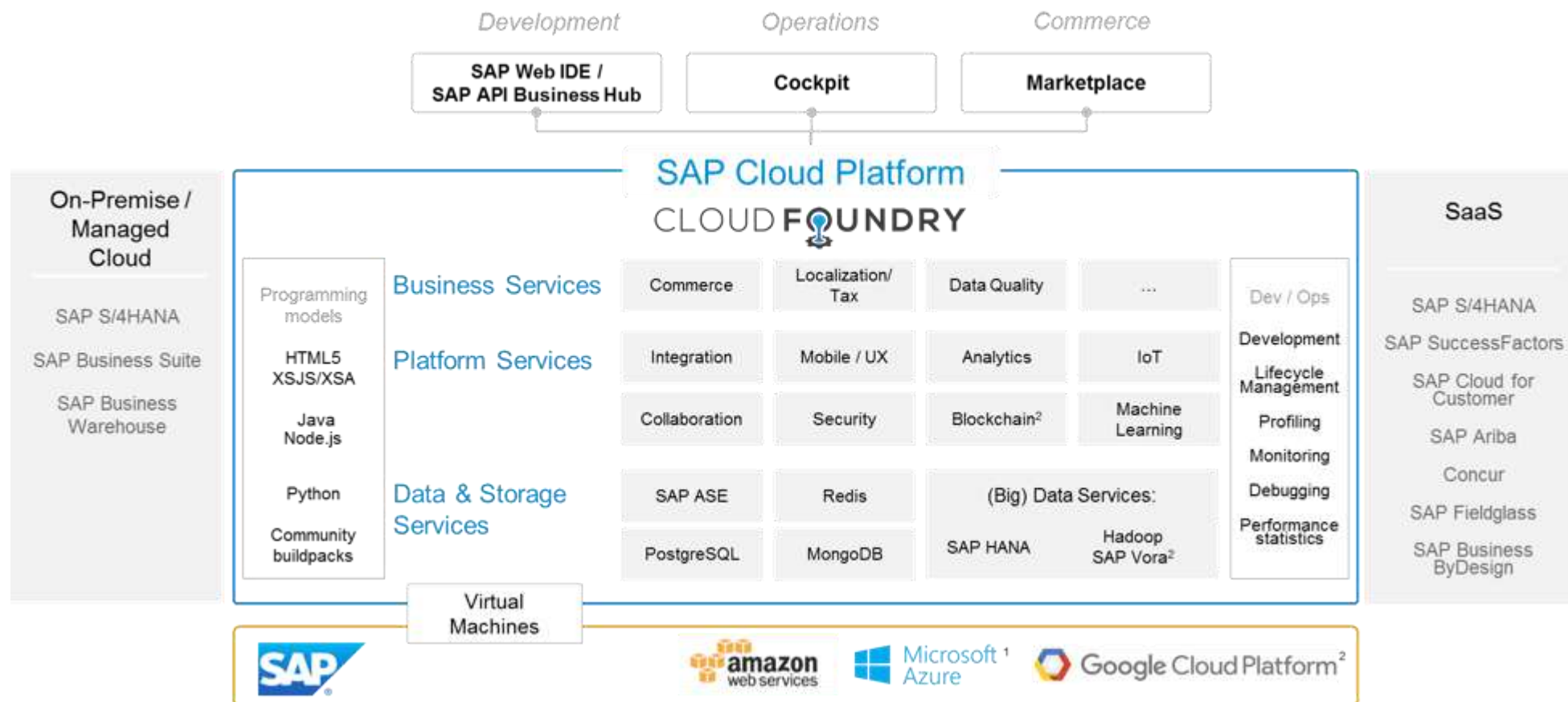


Рисунок 2.2 – Архітектура SCP [8]

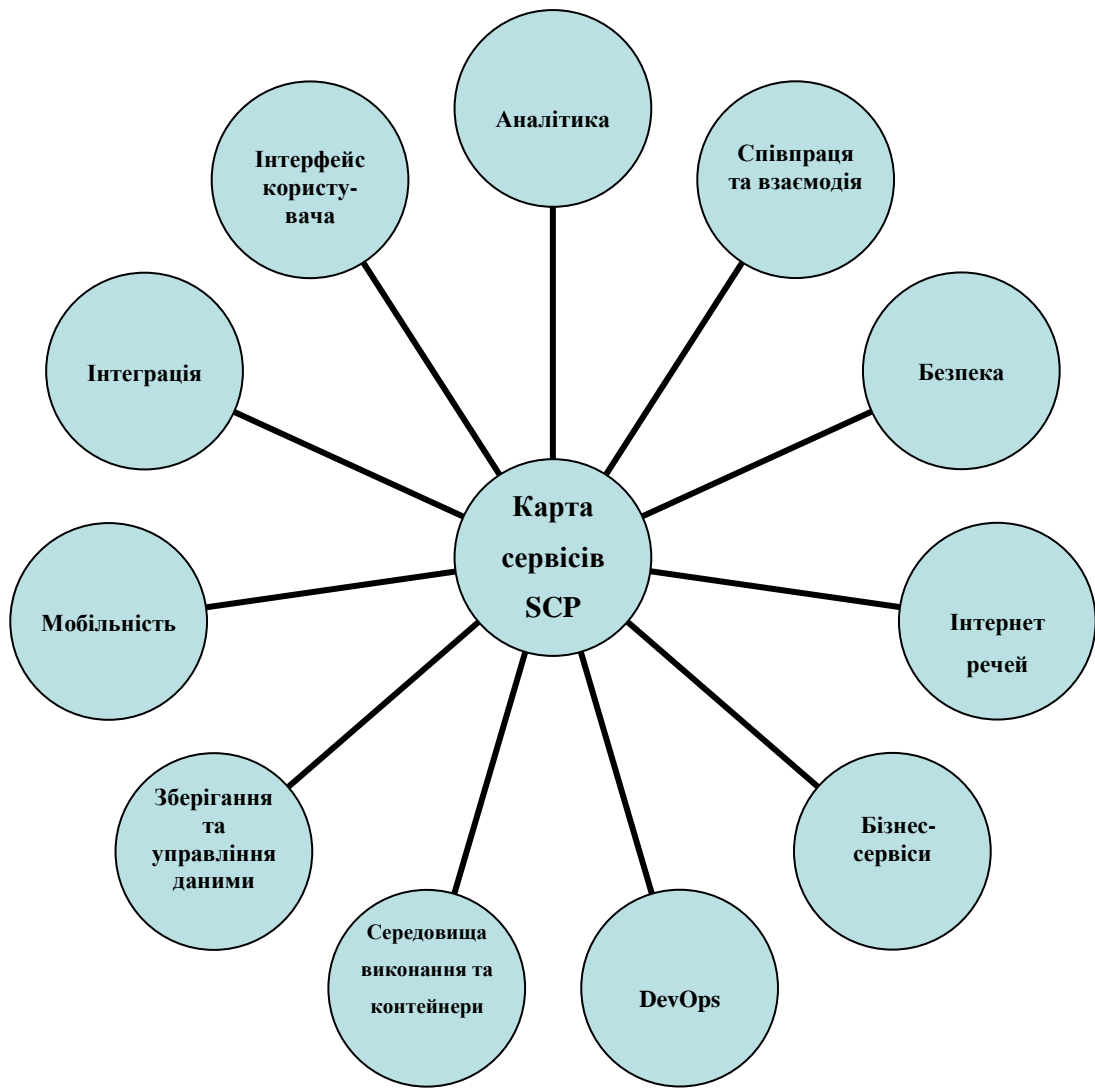


Рисунок 2.3 – Карта сервісів SCP

DevOps позиціонується як Agile-підхід для усунення організаційних і часових бар'єрів між командами розробників та інших учасників життєвого циклу ПЗ (тестувальниками, адміністраторами, тощо), щоб вони могли швидше і надійніше збирати, тестувати і випускати релізи програмних продуктів [9].

DevOps, як і інші Agile-практики, орієнтований на командну роботу, де розглядаються всі аспекти життєвого циклу ПЗ, від програмного коду до експлуатації продукту кінцевим користувачем [10]:



Рисунок 2.4 – Концепція DevOps

- кодування - розробка і аналіз, контроль версій і злиття коду;
- складання - безперервна інтеграція різних складань;
- тестування - безперервне тестування, що забезпечує зворотній зв'язок за бізнес-ризиками;
- упакування - репозиторій артефактів, попередня установка застосунку;
- випуск - управління змінами, офіційне затвердження випуску, автоматизація випуску;
- налаштування - управління інфраструктурою як кодом;
- моніторинг - моніторинг продуктивності застосунків, досвід роботи з кінцевим користувачем.

Оскільки процеси DevOps охоплюють весь цикл поставки ПЗ, виділяють кілька основних цілей цього підходу [10]:

- скорочення часу для виходу на ринок;
- зниження частоти відмов нових релізів;
- скорочення часу виконання виправлень;
- зменшення кількості часу на відновлення при збої нової версії або інших випадках відключення поточної системи.

Ці цілі досягаються через рішення наступних завдань:

- узгодження процесів розробки і постачання ПЗ з експлуатацією;
- автоматизація процесів розробки, тестування і розгортання;
- безперервне тестування якості застосунків;
- управління IT-інфраструктурою як кодом;
- управління змінами;
- безперервний моніторинг продуктивності застосунків і стану інфраструктури.

Фреймворк CALMS (Culture, Automation, Lean, Measurement, Sharing) дозволяє оцінити готовність переходу до DevOps та може застосовуватися як модель для виявлення можливих удосконалень [11].

Розглядаючи DevOps як масштабування Agile-підходу на весь процес розробки, впровадження та супровіду ПЗ, можна виділити 5 основних принципів його реалізації з метою збільшення частоти релізів і підвищення відповідальності команди за продукт (рис. 2.5) [9].

Культура (Culture) крос-функціональне співробітництво різнопрофільних спеціалістів і команд за рахунок єдиного інформаційного простору проєктного контенту, відкритих каналів комунікацій і постійного спілкування всіх учасників.

Автоматизація (Automation) - використання інструментів безперервного постачання з прогоном кожної правки коду через серію автоматизованих тестів, що часто використовують хмарну інфраструктуру, з подальшою упаковкою успішних складань з подальшим переміщенням на робочий сервер за допомогою автоматизованих розгортань і управління інфраструктурою як кодом через конфігурації саморозгортуваних середовищ.

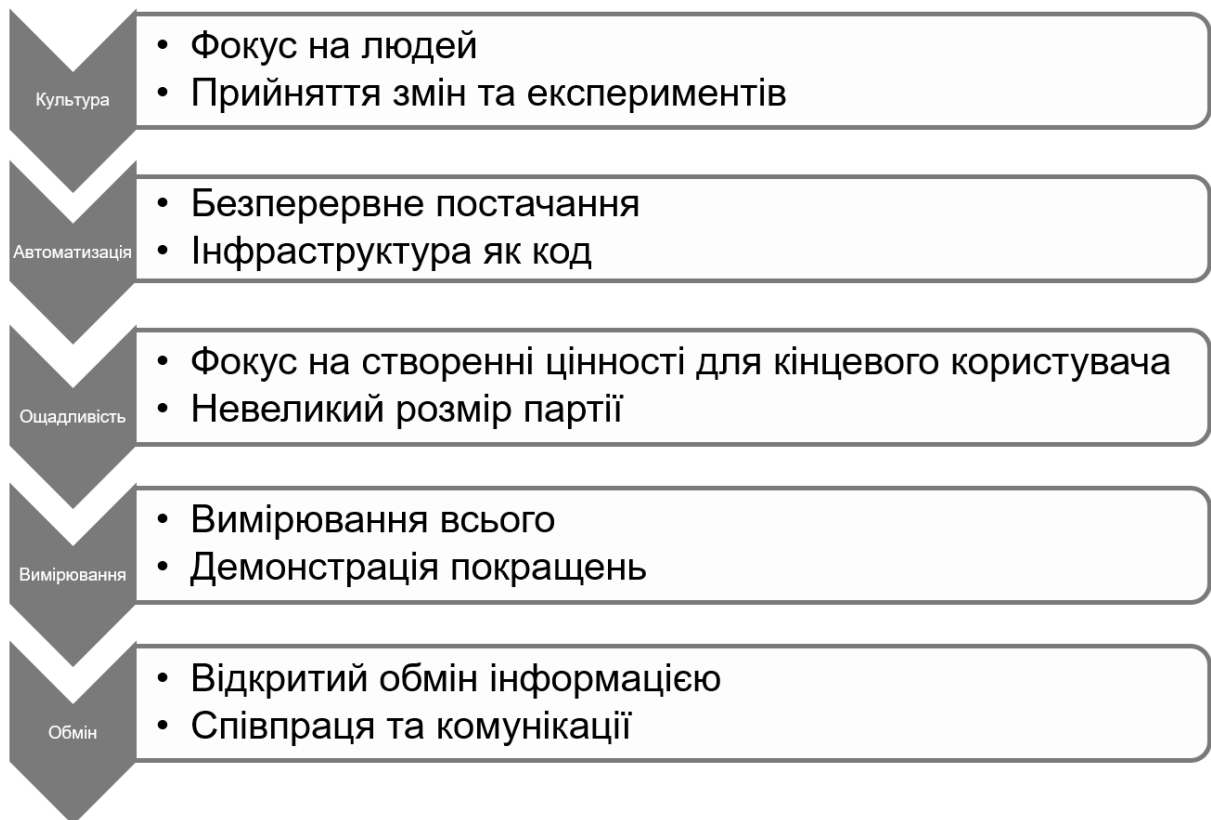


Рисунок 2.5 – Фреймворк CALMS

Ощадливість (Lean) - усунення дій з низькою корисністю і прискорення процесів, безперервне вдосконалення через регулярний ретроспективний аналіз, роздільне тестування різних інструментів, прийняття поразок, можливості швидкого виявлення проблем та їх негайного вирішення.

Вимірювання (Measurement) продуктивності, наприклад, тривалість роботи користувачів з продуктом, частота появи в логах повідомлень про критичні помилки, необхідні ясні і чіткі критерії оцінки роботи, показники ефективності процесів.

Обмін (Sharing) - спільна відповідальність і поділ успіхів, випуск і забезпечення роботи програми здійснюються тими ж людьми, що виконували його збірку, тобто розробники (Developers) і оператори (Operators) взаємодіють на кожному етапі життєвого циклу програми [11].

Завдяки стандартизації та автоматизації процесів розробки і впровадження, DevOps дає наступні переваги в управлінні випуском ПЗ [10]:

- події, документовані процеси управління і докладні звіти легко відстежувати;
- розробники мають більше контролю над середовищем, надаючи інфраструктурі більш прикладне розуміння продукту і процесів його експлуатації;
- значне скорочення часу виходу на ринок за рахунок «безшовного» циклу розробки і впровадження;
- поліпшення задоволеності клієнтів;
- підвищення якості та надійності продукції;
- збільшення продуктивності і ефективності;
- швидкість реагування і висока швидкість експериментів;
- розширення компетенцій і відповідальності розробників - програмісти беруть участь в налаштуванні серверів і пошуку помилок, пишуть автоматизовані тести, згладжуючи можливі інфраструктурні уразливості в коді. Це скорочує кількість помилок при розгортанні застосунків приблизно в 5 разів [12].

Можна виділити також певні недоліки методології.

Неповнота циклу - за бортом процесів DevOps залишаються важливі етапи життя ПЗ, як розробка і аналіз вимог, а також проєктування архітектури. Також можливо упущення ручного тестування, що може бути критично в деяких випадках [12]. Наприклад, якщо розробники недостатньо якісно проаналізували вимоги і протестували продукт, розглядаючи його з точки зору «ідеального» коду, а не з позиції користувачів, результат може бути незручним в експлуатації [13].

Недостатній професіоналізм учасників, які розбираються у всьому (розробка, тестування, розгортання, підтримка), але поверхово.

Високе навантаження на менеджмент - якщо у розробників і операторів немає загальних цілей, в цьому винні менеджери, які не організували ефективну взаємодію між командами різнопрофільних фахівців.

Для вирішення цієї проблеми потрібна нова система оцінки менеджерів на основі відгуків від підлеглих [12].

Методологічно DevOps підтримує принципи Agile і Continuous delivery - безперервного постачання ПЗ. Для організації процесів можуть бути використані такі методи Agile, як Scrum, Kanban і їх варіанти.

У плані інструментів для автоматизації процесів розробки, тестування і розгортання використовуються наступні технології і популярні засоби їх реалізації:

- розподілений контроль версій (Git, Mercurial, Subversion, CVS);
- контейнеризація (Docker, Rocket, Kubernetes);
- безперервна інтеграція - складання та тестування кінцевого продукту (Jenkins, TeamCity, Bamboo);
- управління інфраструктурою як кодом (Puppet, Chef, Ansible);
- віртуалізація (Vagrant);
- балансування хмарних ресурсів (VMware DRS) [14].

DevOps може бути корисний практично будь-якій організації, пов'язаній з розробкою застосунків або управлінням великою кількістю серверів. Великі ІТ-гіганти (Amazon, Adobe, Google, Facebook і т.д.) і технологічні підприємства інших сфер (Netflix, Walmart, Etsy та ін.) потребують DevOps-фахівців. Методологія активно використовується в банківській, телекомунікаційній та ІТ-галузях [14].

## **2.4 Огляд хмарних рішень на основі SAP HANA**

SAP HANA - це система управління базою даних від компанії SAP. Основна перевага системи - швидкодія, яка досягається за рахунок зберігання даних в оперативній пам'яті. За допомогою рішення SAP HANA можна:

- зберігати корпоративні дані в оперативній пам'яті;
- використовувати ефективні алгоритми зберігання та вилучення даних;

– забезпечити розпаралелювання обчислень, використовуючи многоядерну архітектуру рішення.

SAP HANA Cloud Platform (SHCP) - це хмарне рішення, що побудоване на базі SAP HANA і володіє всіма перевагами системи нового покоління (рис.2.6).

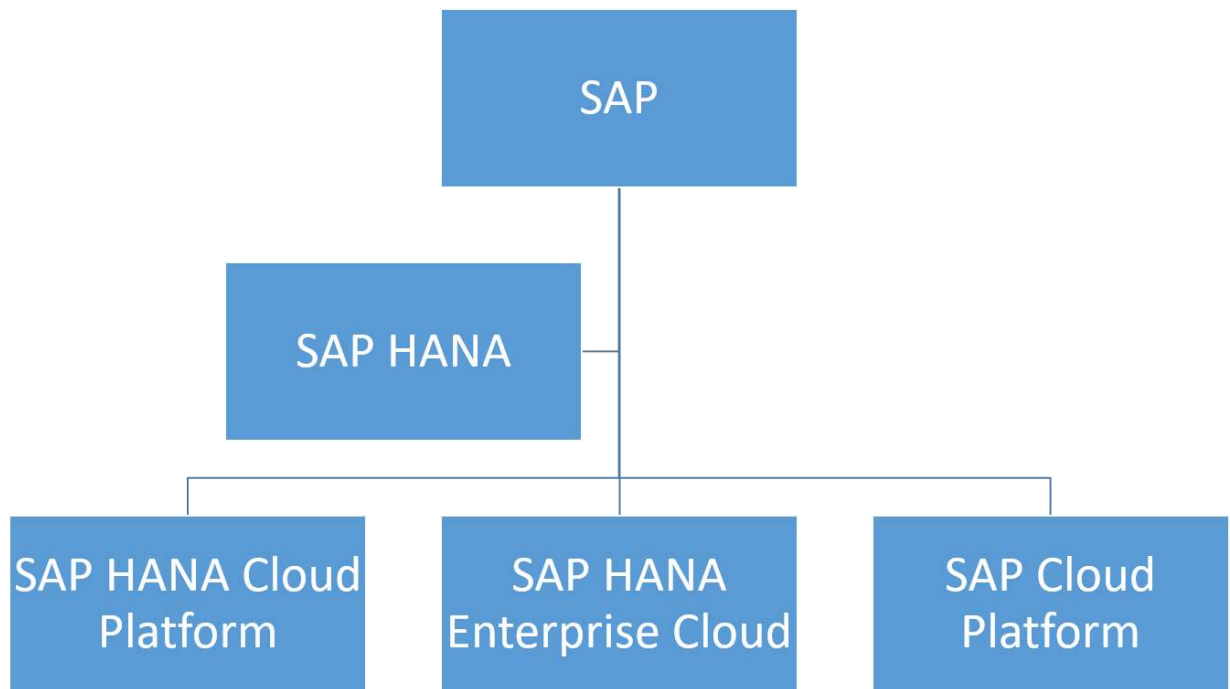


Рисунок 2.6 – Хмарні рішення SAP

SHCP допомагає істотно скоротити витрати при зберіганні і обробці великих масивів даних. Впровадження хмарного рішення не вимагає значних витрат на інфраструктуру - доступ до універсальної платформи SAP HANA здійснюється через дата-центри провайдера і оформляється на умовах передплати. Рішення працює за принципом PaaS.

За допомогою SHCP можна побудувати гібридні рішення управління підприємством, коли частина ресурсів знаходиться на території клієнта, а інша - в «хмарі». Така гнучка система роботи з даними забезпечує індивідуальний підхід до кожної компанії: замовник самостійно визначає

терміни і масштаб розгортання системи, а також має можливість подальшого розширення функціональності власними силами.

На базі SHPC створено рішення SAP HANA Enterprise Cloud (SHEC), яке працює за принципом IaaS. Підприємство отримує доступ до платформи і інфраструктурних рішень на базі SAP HANA, в тому числі до SAP S / 4 HANA - наступного покоління системи SAP Business Suite [15].

Платформи SCP та SHEC мають різне призначення і мету. Обидві ці платформи є варіаціями хмарної технології HANA, але використовують різні сервісні моделі. У той час як SCP пропонує інструмент по моделі PaaS, призначений для розробки та запуску хмарних застосунків, SHEC являє собою інструмент IaaS, який дозволяє компаніям виконувати операції на основі SAP в розміщеному середовищі.

SAP розміщує застосунки SHEC в різних центрах обробки даних, розташованих по всьому світу, і забезпечує постійну підтримку і управління застосунками, які включають оновлення, виправлення, резервні копії, відновлення та відновлення, інфраструктуру і моніторинг, а також виявлення інцидентів [7].

SHEC - це приватна служба хмарного хостингу. Хмарний сервіс дозволяє клієнтам і партнерам розгорнути застосунки HANA, SAP ERP, CRM, SAP BW та інші в своїй приватній хмарі, як повністю управляється і підтримується SAP. В результаті є можливість скоротити витрати на створення інфраструктури.

У хмарному сервісі реалізовано програмне забезпечення SAP Business Suite, яке включає в себе замовні і готові програми SAP HANA і ПЗ SAP NetWeaver Business Warehouse на одному екземплярі платформи бази даних SAP HANA в пам'яті.

Згідно SAP, ключовими перевагами SHEC і SAP HANA в цілому є те, що транзакційні й аналітичні дані можуть оброблятися з більшою швидкістю в порівнянні з традиційними платформами баз даних.

SAP робить SHEC доступним через центри обробки даних по всьому світу. Керовані служби SHEC включають резервне копіювання, виправлення, надання та оновлення, відновлення і відновлення, моніторинг інфраструктури та виявлення подій - все це в приватному хмарному середовищі [16].

SHEC - це приватна хмара, відповідно, всі призначені для користувача ландшафти ізольовані один від одного, і адміністрування здійснюється за допомогою захищеного мережевого управління. В даний час SHEC підтримує роботу таких програм компанії SAP як: ECC 6, CRM 7.0, BW 7.3, SAP HANA DW / DM.

Однак платформа SHEC призначена не тільки для хостингу бізнес-рішень SAP. Вона може використовуватись і як PaaS сервіс, за допомогою якого розробники отримують можливість створювати високо-продуктивні корпоративні застосунки. Економічна ефективність SHEC в порівнянні з локально використовуваним апаратним і програмним забезпеченням полягає в суттєвому скороченні витрат на апаратно-програмну інфраструктуру. При цьому очевидно, що бізнес вкрай обережно ставиться до будь-якої модернізації, будь-якої зміни розгорнутої на підприємстві програмно-апаратної інфраструктури, через страх завдати шкоди основній діяльності, яку можуть спричинити, наприклад, застій інформаційної системи або втрата даних. У моделі SHEC подібні ризики відсутні. Крім того, ітеративний процес модернізації платформи в «хмарі» проходить непомітно для кінцевих користувачів, що дозволяє користувачам негайно почати користуватися новими функціональними можливостями.

Серед основних економічних переваг SHEC необхідно виділити наступні:

- низькі бар'єри переходу в in-memo;у;
- гнучкі умови ліцензування та відсутність капітальних інвестицій в інфраструктуру;
- максимальна віддача інвестицій.

Економічна ефективність SHEC тісно пов'язана з технологічними перевагами, в їх числі:

- підвищення швидкості роботи бізнес-застосунків;
- швидке розгортання;
- хостинг інфраструктури і повна підтримка від SAP;
- додаткові послуги з оцінки, адміністрування бізнес-застосунків, замовні розробки;
- глобальні центри обробки даних розгортаються підтримуються на спеціалізованих платформах під управлінням професійних фахівців, що визначає високий рівень відмовостійкості та надійності системи і забезпечує необхідну безпеку [17].

На базі універсальної платформи SAP HANA створено також SAP S / 4 HANA. Це система управління підприємством, фактично подальший розвиток системи SAP Business Suite. У рішенні нового покоління стандартні транзакції оптимізовані під роботу з даними в оперативній пам'яті, що забезпечує багаторазове прискорення за часом обробки даних. У випадку з «важкими» транзакціями, які вимагають обробки великого обсягу даних, прискорення може становити кілька тисяч разів.

Інтуїтивно зрозумілий інтерфейс рішення, побудований на базі SAP Fiori, спрощує роботу з даними і використання систем, що дозволяє збільшити ефективність роботи з системою SAP. Рішення SAP Simple Finance істотно розширює функціональність стандартного модуля FI, в першу чергу за рахунок широких аналітичних можливостей.

Таким чином, використання рішення SAP S / 4 HANA має наступні переваги:

- збільшення продуктивності системи, особливо для транзакцій, які потребують обробки великого обсягу даних;
- збереження високої продуктивності в період пікових навантажень;

- можливість запуску транзакцій без обмежень на обсяг вибірки в режимі онлайн;
- можливість багаторазового збільшення кількості запусків транзакцій відповідно до реальної необхідності.

На платформі SAP HANA також функціонують інші рішення SAP, наприклад, продукти лінійки SAP Business Objects Business Intelligence і інші хмарні рішення [18].

## **2.5 Висновки до розділу 2**

Як показали проведені дослідження, SAP Cloud Platform відноситься до хмарних платформ, що забезпечують середовище виконання для розробки, тестування та управління застосунками. Одним з сервісів SCP є DevOps - набір практик для підвищення ефективності процесів розробки і експлуатації ПЗ, що підтримує принципи Agile і Continuous delivery - безперервного постачання ПЗ. Фреймворк CALMS дозволяє оцінити готовність переходу до DevOps та може застосовуватися як модель для виявлення можливих удосконалень процесу розробки, впровадження та супровіду ПЗ. Важливою особливістю SCP є також підтримка так званої моделі BYOL, що забезпечує використання будь-якої мови програмування, наприклад, PHP, Python.

## 3 РОЗРОБКА БІЗНЕС-ЗАСТОСУНКУ НА SAP CLOUD PLATFORM

### 3.1 Аналіз технічного завдання

Розроблюваний продукт має назву «Наказ про переведення», відноситься до функціонального напрямку «Адміністрування персоналу» та призначений для формування звітнього документа «Наказ про переведення» за заданим шаблоном в форматі Word при натисненні кнопки «Печать» IT0298 «Накази». Мова формування звіту – українська.

Необхідно розробити програму друку ZBHR\_PRINT\_03OR та нову друковану форму ZBHR\_03OR – «Наказ про переведення» та виконати налаштування зв'язку цієї форми та виду заходу 02 «Переведення».

При розробці звіту необхідно використовувати базу даних PNPCE. Всі дані необхідно аналізувати на дату початку нового організаційного присвоєння (P0001-BEGDA). Дані звітнього документа наведено в таблиці 3.1.

Таблиця 3.1 Опис формування даних звіту

Ідентифікатор значення	Опис формування даних
1	2
A1	Статичне значення «Наказ про переведення на іншу роботу»
A2	Номер наказу
A3	Дата наказу
A4	Табельний номер працівника
A5	Прізвище, ім'я, по батькові працівника в називному відмінку
A6	Текст залежить від причини заходу «Переведення». Для причини заходу: «01 Переведення на іншу посаду безстроково» виводиться «На невизначений час». Для причини заходу: «02 Переведення на іншу посаду тимчасово» виводиться текст уточнення причини заходу: 01 «На час відсутності основного працівника»; 02 « За строковим трудовим договором на період відсутності основного працівника»; 03 « На час виконання певної роботи» 04 « тимчасово»

Продовження табл. 3.1

1	2
A7	Назва попередньої штатної посади співробітника (не штатної посади) у називному відмінку
A8	Назва нової штатної посади співробітника у називному відмінку
A9	Назва попереднього розряду співробітника (не штатної посади) у називному відмінку
A10	Код та Назва попередньої категорії працівника у називному відмінку
A11	Назва нового розряду співробітника у називному відмінку
A12	Код та Назва нової категорії працівника у називному відмінку
A13	Назва підрозділу, з якого переводиться співробітник (в називному відмінку). Відображається ієрархія підрозділів від рівня цеху/відділу/управління (1-й підрозділ після ЗАЕС) до безпосереднього підрозділу
A14	Назва підрозділу, в який переводиться співробітник (в називному відмінку). Відображається ієрархія підрозділів від рівня цеху/відділу/управління (перший підрозділ після ЗАЕС) до безпосереднього підрозділу
A15	Виводиться попередній Посадовий оклад співробітника (ІТ0008), КВО 1000
A16	Виводиться попередня назва, сума доплати для співробітника, в разі наявності
A17	Виводиться попередня назва, сума надбавки для співробітника, в разі наявності
A18	Виводиться новий Посадовий оклад співробітника (ІТ0008), КВО 1000
A19	Виводиться нова назва, сума доплати для співробітника, в разі наявності
A20	Виводиться нова назва, сума надбавки для співробітника, в разі наявності
A21	Визначення форми і система оплати праці. Статичне значення «почасово-преміальна»
A22	Визначення умов праці за попередньою посадою: «НУ» для 0 Нормальні умови праці; «ШУ» для 1 Шкідливі умови праці; «ВУ» для 2 Важкі умови праці; «ОШУ» для 3 Особливо шкідливі умови праці; «ОВУ» для 4 Особливо важкі умови праці

Продовження табл. 3.1

1	2
A23	Виведення ознаки віднесення посади до Списку за попередньою посадою
A24	Визначення умов праці за новою посадою: «НУ» для 0 Нормальні умови праці; «ШУ» для 1 Шкідливі умови праці; «ВУ» для 2 Важкі умови праці; «ОШУ» для 3 Особливо шкідливі умови праці; «ОВУ» для 4 Особливо важкі умови праці
A25	Виведення ознаки віднесення посади до Списку за новою посадою
A26	Виведення назви графіку роботи із ІТ0007
A27	Виведення часу початку робочого дня у форматі «гг.хх». Виводиться тільки для нормальних (незмінних) графіків. Ведеться вручну ІТ0298 Накази
A28	Виведення часу закінчення робочого дня у форматі «гг.хх». Виводиться тільки для нормальних (незмінних) графіків. Ведеться вручну ІТ0298 Накази
A29	Виведення часу початку та закінчення перерви у форматі «гг.хх – гг.хх». Виводиться тільки для нормальних (незмінних) графіків. Ведеться вручну ІТ0298 Накази
A30	Заповнюється вручну
A31	Текст для підстави наказу. Текст буде вводиться вручну в ІТ0298 «Накази» по кнопці «Підстава для наказу»
A32	Штатна посада підписанта наказу
A33	Ініціали та прізвище підписанта наказу
A34, A36, A38, A40	Штатна посада особи, яка погоджує наказ, вказується вручну
A35, A37, A39, A41	Ініціали та прізвище особи, яка погоджує наказ, вказується вручну
A42	Ініціали та прізвище працівника в називному відмінку, якого переведено (ознайомлення) вказується вручну
A43	Ініціали та прізвище працівника УК (виконавця) вказується вручну
A44	Прізвище, ім'я, по батькові співробітника в називному відмінку
A45	Табельний номер співробітника

Продовження табл. 3.1

1	2
A46	Назва підрозділу, з якого переводиться співробітник (в називному відмінку). Відображається ієрархія підрозділів від рівня цеху/відділу/управління (перший підрозділ після ЗАЕС) до безпосереднього підрозділу . Назва штатної посади в називному відмінку.
A47	Назва підрозділу, в який переводиться співробітник (в називному відмінку). Відображається ієрархія підрозділів від рівня цеху/відділу/управління (перший підрозділ після ЗАЕС) до безпосереднього підрозділу . Назва штатної посади в називному відмінку.
A48	Ознака суб'єкта декларування на посаді з якої переводиться (IT1010)
A49	Ознака суб'єкта декларування на посаді на яку переводиться (IT1010)

Заповнення друкованої форми відбувається наступним чином.

[A0] Назва організації. Для P0001-BUKRS = 0100 виводити значення: ВП «Запорізька АЕС». Для P0001-BUKRS = NAEK виводити значення: «Дирекція».

[A0.1] Код ЄРДПОУ організації. Для P0001-BUKRS = 0100 виводити значення: 19355964. Для P0001-BUKRS = NAEK виводити значення: 22222222.

[A2] Номер наказу. Заповнюється значенням P0298-ORDNU. Таблиця P0298 заповнюється стандартними засобами ЛБД PNPCE.

[A3] Дата наказу. Заповнюється значенням P0298-ORDDT. Таблиця P0298 заповнюється стандартними засобами ЛБД PNPCE.

[A4] Табельний номер працівника. Виводиться поле PERNR-PERNR.

[A5] Прізвище, ім'я, по батькові співробітника в називному відмінку. Заповнюється значенням P0002-ZB\_NACHNZ, P0002-ZB\_VORNAZ, P0002-ZB\_MIDNMZ. Таблиця P0002 заповнюється стандартними засобами ЛБД PNPCE.

[A6] Термін переведення. Якщо P0298-MASSG=01, то виводиться текст: «На невизначений час». В інших випадках, виводиться довгий текст

уточнення причини заходу за допомогою ФМ HR\_RU\_READ\_MASS\_TEXT (SPRSL=UA, MASSN=P0298-MASSN, MASSG=P0298-MASSG, FR530="пусто", MASSS=P0298-MASSS) виводимо текст із поля MLTXT\_LONG.

[A7] Назва штатної посади, з якої переводять співробітника, у називному відмінку. Визначаємо по ID штатної посади (P0001-PLANS на дату BEGDA, ENDDA = P0298-BEGDA мінус 1 день) за допомогою ФМ RH\_OBJECT\_DESCRIPTION\_READ (SUBTY = «9001», OTYPE = «S», OBJID = P0001-PLANS) текст з поля TLINE.

[A8] Назва штатної посади, на яку переводять співробітника, у називному відмінку. Визначаємо по ID штатної посади (P0001-PLANS) за допомогою ФМ RH\_OBJECT\_DESCRIPTION\_READ (SUBTY = «9001», OTYPE = «S», OBJID = P0001-PLANS) текст з поля TLINE.

[A9] Назва попереднього розряду співробітника у називному відмінку. Вивести текстове значення (GS\_BIT9005\_QUAL-QUALIF\_TEXT) для коду в полі P9005-QUALIFICATION (ФМ HR\_READ\_INFOTYPE на дату BEGDA, ENDDA = P0298-BEGDA мінус 1 день).

[A10] Назва попередньої категорії по роду роботи співробітника у називному відмінку. Вивести код та текстове значення з поля P0001-PERSK (ФМ HR\_READ\_INFOTYPE на дату BEGDA, ENDDA = P0298-BEGDA мінус 1 день).

[A11] Назва нового розряду співробітника у називному відмінку. Вивести текстове значення (GS\_BIT9005\_QUAL-QUALIF\_TEXT) для коду в полі P9005-QUALIFICATION (ФМ HR\_READ\_INFOTYPE).

[A12] Назва нової категорії по роду роботи співробітника у називному відмінку. Вивести код та текстове значення з поля P0001-PERSK (ФМ HR\_READ\_INFOTYPE).

[A12.1] Приналежність персоналу з якої переводиться. Текст для групи працівників. Необхідно прочитати значення P0001-WERKS (ФМ HR\_READ\_INFOTYPE на дату BEGDA, ENDDA = P0298-BEGDA мінус 1

день): Якщо P0001-WERKS=0101, NA01 то виводиться текст: «ПВП»; якщо P0001-WERKS=0102, NA02, то виводиться текст: «НП».

[A13] Назва підрозділу, з якого переводиться співробітник, у називному відмінку. По ID Організаційної одиниці (P0001-ORGEN (ФМ HR\_READ\_INFOTYPE на дату BEGDA, ENDDA = P0298-BEGDA мінус 1 день)) будується ієрархія організаційних одиниць RESULT\_STRUC за допомогою ФМ RH\_STRUC\_GET, шлях аналізу «A002». Отримана структура сортується по полю LEVEL за спаданням. Для кожної організаційної одиниці визначається повне найменування підрозділу в називному відмінку за допомогою ФМ RH\_OBJECT\_DESCRIPTION\_READ (SUBTY = «9001», OTYPE = «O», OBJID = RESULT\_STRUC-OBJID) та вивести в порядку спадання з абзацу (два останніх (перших) рівня не виводяться) (RESULT\_STRUC-LEVEL текст з поля TLINE.

[A13.1] Приналежність персоналу куди переводиться. Текст для групи працівників. Необхідно прочитати значення P0001-WERKS (ФМ HR\_READ\_INFOTYPE): Якщо P0001-WERKS=0101, NA01 то виводиться текст: «ПВП»; якщо P0001-WERKS=0102, NA02, то виводиться текст: «НП».

[A14] Назва підрозділу, в який переводиться співробітник, у називному відмінку. По ID Організаційної одиниці (P0001-ORGEN (ФМ HR\_READ\_INFOTYPE)) будується ієрархія організаційних одиниць RESULT\_STRUC за допомогою ФМ RH\_STRUC\_GET, шлях аналізу «A002». Отримана структура сортується по полю LEVEL за спаданням. Для кожної організаційної одиниці визначається повне найменування підрозділу в називному відмінку за допомогою ФМ RH\_OBJECT\_DESCRIPTION\_READ (SUBTY = «9001», OTYPE = «O», OBJID = RESULT\_STRUC-OBJID) та вивести в порядку спадання з абзацу (два останніх (перших) рівня не виводяться) (RESULT\_STRUC-LEVEL текст з поля TLINE.

[A15] Посадовий оклад співробітника, попередній. Необхідно прочитати та вивести значення P0008-BET\* для P0008-LGA\*= «1000» на дату

P0298-BEGDA мінус 1 день. Таблиця P0008 заповнюється стандартними засобами ЛБД PNPCE.

[A16] Доплата попередня. За допомогою ФМ RP\_FILL\_WAGE\_TYPE\_TABLE\_EXT (BEGDA=P0298-BEGDA мінус 1 день; PERNR=PERNR), прочитати результат в PPBWLА: необхідно прочитати значення LGAR; для LGAR = виводиться текст: «надбавку за розширення зони відповідальності в розмірі P0008-BET\* грн. з P0008-BEGDA по P0008-ENDDA».

[A17] Надбавка попередня. За допомогою ФМ RP\_FILL\_WAGE\_TYPE\_TABLE\_EXT (BEGDA=P0298-BEGDA мінус 1 день; PERNR=PERNR), прочитати результат в PPBWLА: необхідно прочитати значення LGAR; для LGAR = виводиться текст: «надбавку за розширення зони відповідальності в розмірі P0008-BET\* грн. з P0008-BEGDA по P0008-ENDDA».

[A18] Посадовий оклад співробітника, новий. Необхідно прочитати та вивести значення P0008-BET\* для P0008-LGA\*= «1000». Таблиця P0008 заповнюється стандартними засобами ЛБД PNPCE.

[A19] Доплата нова. За допомогою ФМ RP\_FILL\_WAGE\_TYPE\_TABLE\_EXT (BEGDA=P0298-BEGDA; PERNR=PERNR), прочитати результат в PPBWLА: необхідно прочитати значення LGAR; для LGAR = виводиться текст: «надбавку за розширення зони відповідальності в розмірі P0008-BET\* грн. з P0008-BEGDA по P0008-ENDDA».

[A20] Надбавка нова. За допомогою ФМ RP\_FILL\_WAGE\_TYPE\_TABLE\_EXT (BEGDA=P0298-BEGDA; PERNR=PERNR), прочитати результат в PPBWLА: необхідно прочитати значення LGAR; для LGAR = виводиться текст: «надбавку за розширення зони відповідальності в розмірі P0008-BET\* грн. з P0008-BEGDA по P0008-ENDDA».

[A21] Форма і система оплати праці. Статичне значення «Почасово-преміальна».

[A22] Умови праці за попередньою посадою. За допомогою ФМ RH\_READ\_INFITY\_1001 (PLVAR=01; OTYPE=S; OBJID= P0001-PLANS (для PERNR); SUBTY=B003; BEGDA= P0298-BEGDA мінус 1 день) визначити код з поля HRP1001-SOBID для HRP1001-SCLAS=A.

За допомогою ФМ RH\_READ\_INFITY (PLVAR=01; OTYPE=A; OBJID= HRP1001-SOBID (для HRP1001-SCLAS=A); INTY=1640; BEGDA= P0298-BEGDA мінус 1 день) визначити та вивести відповідне значення для коду з поля P1640-LBCOND:

P1640-LBCOND=0 вивести «НУ»;

P1640-LBCOND=1 вивести «ШУ»;

P1640-LBCOND=2 вивести «ВУ»;

P1640-LBCOND=3 вивести «ОШУ»;

P1640-LBCOND=4 вивести «ОВУ»;

[A23] Список, код за попередньою посадою. За допомогою ФМ RH\_READ\_INFITY (PLVAR=01; OTYPE=A; OBJID= HRP1001-SOBID (для HRP1001-SCLAS=A) [A16]; INTY=1640; BEGDA= P0298-BEGDA мінус один день) визначити та вивести відповідне значення для коду з полів через кому P1640-LSTNUM, P1640-ZLSTNUM;

[A24] Умови праці за новою посадою. За допомогою ФМ RH\_READ\_INFITY\_1001 (PLVAR=01; OTYPE=S; OBJID= P0001-PLANS (для PERNR); SUBTY=B003; BEGDA= P0298-BEGDA) визначити код з поля HRP1001-SOBID для HRP1001-SCLAS=A.

За допомогою ФМ RH\_READ\_INFITY (PLVAR=01; OTYPE=A; OBJID= HRP1001-SOBID (для HRP1001-SCLAS=A); INTY=1640; BEGDA= P0298-BEGDA мінус 1 день) визначити та вивести відповідне значення для коду з поля P1640-LBCOND:

P1640-LBCOND=0 вивести «НУ»;

P1640-LBCOND=1 вивести «ШУ»;

P1640-LBCOND=2 вивести «ВУ»;

P1640-LBCOND=3 вивести «ОШУ»;

P1640-LBCOND=4 вивести «ОВУ»;

[A25] Список, код за новою посадою. За допомогою ФМ RH\_READ\_INFITY (PLVAR=01; OTYPE=A; OBJID= HRP1001-SOBID (для HRP1001-SCLAS=A) [A16]; INTY=1640; BEGDA= P0298-BEGDA) визначити та вивести відповідне значення для коду з полів через кому P1640-LSTNUM, P1640-ZLSTNUM;

[A26] № нового графіку роботи. Виводити значення з поля P0007-SCHKZ (ФМ RH\_READ\_INFITY) та V\_T508A-RTEXT (для P0007-SCHKZ=V\_T508A-SCHKZ). Таблиця P0007 заповнюється стандартними засобами ЛБД PNPCE.

[A27] Час початку робочого дня

[A28] Час закінчення робочого дня

[A29] Час початку та закінчення перерви

[A30] Індивідуальні умови. Статичне значення «Немає»

[A31] Текст підстави наказу. Заповнити значенням поля текст «Підстава для наказу» IT0298 (за допомогою ФМ READ\_TEXT). Таблиця P0298 заповнюється стандартними засобами ЛБД PNPCE. Для PERNR визначаємо ідентифікатор [NAME] шляхом зклеювання коду в полі P0298-LGTXT та константи зі значенням 0 (Підстава для наказу).

За допомогою ФМ READ\_TEXT (ID=TEXT, LANGUAGE=UA, NAME=[NAME], OBJECT=HRITTXТ) знаходимо таблицю значень та виводимо всі рядки як один.

[A32] Назва штатної посади підписанта наказу у називному відмінку. Для табельного номера керівника (P0298-CHPER) необхідно визначити штатну посаду (P0001-PLANS) за допомогою ФМ HR\_READ\_INFOTYPE.

За ID штатної посади (P0001-PLANS) за допомогою ФМ RH\_OBJECT\_DESCRIPTION\_READ (SUBTY = «0001», OTYPE = «S», OBJID = P0001-PLANS) виводимо текст з поля TLINE.

[A33] Ініціали та прізвище підписанта наказу у називному відмінку. Для табельного номера керівника (P0298-CHPER) знаходимо та виводимо значення з P0002-INITS, P0002-NACHN. Таблиця P0002 заповнюється стандартними засобами ЛБД PNPCE.

[A34], [A36], [A38], [A40] Штатна посада особи, яка погоджує наказ у називному відмінку. Для табельного номера керівника (P0298-ZCHPER2, P0298-ZCHPER3, P0298-ZCHPER4, P0298-ZCHPER5) необхідно визначити штатну посаду (P0001-PLANS) за допомогою ФМ HR\_READ\_INFOTYPE.

За ID штатної посади (P0001-PLANS) за допомогою ФМ RH\_OBJECT\_DESCRIPTION\_READ (SUBTY = «0001», OTYPE = «S», OBJID = P0001-PLANS) виводимо текст з поля TLINE.

[A35], [A37], [A39], [A41] Ініціали та прізвище особи, яка погоджує наказ у називному відмінку. Для табельного номера керівника (P0298-ZCHPER2, P0298-ZCHPER3, P0298-ZCHPER4, P0298-ZCHPER5) знаходимо та виводимо значення з P0002-INITS, P0002-NACHN. Таблиця P0002 заповнюється стандартними засобами ЛБД PNPCE.

[A42] Ініціали та прізвище працівника в називному відмінку, якого переведено (ознайомлення). Для PERNR виводимо значення з P0002-INITS, P0002-NACHN (ФМ RH\_READ\_INFOTYPE). Таблиця P0002 заповнюється стандартними засобами ЛБД PNPCE.

[A43] Ініціали та прізвище працівника УК (виконавця). Для табельного номера керівника (P0298-ZCHPER6) знаходимо та виводимо значення з P0002-INITS, P0002-NACHN. Таблиця P0002 заповнюється стандартними засобами ЛБД PNPCE.

[A44] Прізвище, ім'я, по батькові співробітника в називному відмінку. Заповнюється значенням P0002-NACHN, P0002-VORNA, P0002-MIDNM (ФМ RH\_READ\_INFOTYPE). Таблиця P0002 заповнюється стандартними засобами ЛБД PNPCE.

[A45] Табельний номер співробітника. Виводиться поле PERNR-PERNR

[A46.1, A46.2, A46.3, A46.4] Назва підрозділу, з якого переводиться співробітник (в називному відмінку). Назва посади:

A46.1 - [A13.1] Приналежність персоналу. Текст для групи працівників. Необхідно прочитати значення P0001-WERKS (ФМ HR\_READ\_INFOTYPE на дату BEGDA, ENDDA = P0298-BEGDA мінус 1 день): Якщо P0001-WERKS=0101, NA01 то виводиться текст: «ПВП»; якщо P0001-WERKS=0102, NA02, то виводиться текст:«НП»

A46.2 - [A13] Назва підрозділу, з якого переводиться співробітник , у називному відмінку. За ID Організаційної одиниці (P0001-ORGEN (ФМ HR\_READ\_INFOTYPE на дату BEGDA, ENDDA = P0298-BEGDA мінус 1 день)) будується ієрархія організаційних одиниць RESULT\_STRUC за допомогою ФМ RH\_STRUC\_GET, шлях аналізу «A002». Отримана структура сортується по полю LEVEL за спаданням.

Для кожної організаційної одиниці визначається повне найменування підрозділу в називному відмінку за допомогою ФМ RH\_OBJECT\_DESCRIPTION\_READ (SUBTY = «9001», OTYPE = «O», OBJID = RESULT\_STRUC-OBJID) та вивести в порядку спадання з абзацу (два останніх (перших) рівня не виводяться) (RESULT\_STRUC-LEVEL текст з поля TLINE.

A46.3 - [A7] Назва штатної посади, з якої переводять співробітника, у називному відмінку. Визначаємо по ID штатної посади (P0001-PLANS на дату BEGDA, ENDDA = P0298-BEGDA мінус 1 день) за допомогою ФМ RH\_OBJECT\_DESCRIPTION\_READ (SUBTY = «9001», OTYPE = «S», OBJID = P0001-PLANS) текст з поля TLINE.

A46.4 - [A22] Умови праці за попередньою посадою. За допомогою ФМ RH\_READ\_INFOTY\_1001 (PLVAR=01; OTYPE=S; OBJID= P0001-PLANS (для PERNR); SUBTY=B003; BEGDA= P0298-BEGDA мінус 1 день) визначити код з поля HRP1001-SOBID для HRP1001-SCLAS=A

За допомогою ФМ RH\_READ\_INFOTY (PLVAR=01; OTYPE=A; OBJID= HRP1001-SOBID (для HRP1001-SCLAS=A); INTY=1640; BEGDA=

P0298-BEGDA мінус 1 день) визначити та вивести відповідне значення для коду з поля P1640-LBCOND:

P1640-LBCOND=0 вивести «НУ»;

P1640-LBCOND=1 вивести «ШУ»;

P1640-LBCOND=2 вивести «ВУ»;

P1640-LBCOND=3 вивести «ОШУ»;

P1640-LBCOND=4 вивести «ОВУ»;

[A47.1, A47.2, A47.3, A47.4] Назва підрозділу на яку переводиться співробітник (в називному відмінку). Назва посади.

A47.1 - [A14.1] Приналежність персоналу. Текст для групи працівників. Необхідно прочитати значення P0001-WERKS (ФМ HR\_READ\_INFOTYPE на дату BEGDA, ENDDA = P0298-BEGDA мінус 1 день): якщо P0001-WERKS=0101, NA01 то виводиться текст: «ПВП»; якщо P0001-WERKS=0102, NA02, то виводиться текст: «НП»

A47.2 - [A14] Назва підрозділу, в який переводиться співробітник, у називному відмінку. за ID Організаційної одиниці (P0001-ORGEN (ФМ HR\_READ\_INFOTYPE)) будується ієрархія організаційних одиниць RESULT\_STRUC за допомогою ФМ RH\_STRUC\_GET, шлях аналізу «A002». Отримана структура сортується по полю LEVEL за спаданням.

Для кожної організаційної одиниці визначається повне найменування підрозділу в називному відмінку за допомогою ФМ RH\_OBJECT\_DESCRIPTION\_READ (SUBTY = «9001», OTYPE = «O», OBJID = RESULT\_STRUC-OBJID) та вивести в порядку спадання з абзацу (два останніх (перших) рівня не виводяться) (RESULT\_STRUC-LEVEL текст з поля TLINE.

A47.3 - [A8] Назва штатної посади, на яку переводять співробітника, у називному відмінку. Визначаємо по ID штатної посади (P0001-PLANS) за допомогою ФМ RH\_OBJECT\_DESCRIPTION\_READ (SUBTY = «9001», OTYPE = «S», OBJID = P0001-PLANS) текст з поля TLINE.

A47.4 - [A24] Умови праці за новою посадою. За допомогою ФМ RH\_READ\_INFITY\_1001 (PLVAR=01; OTYPE=S; OBJID= P0001-PLANS (для PERNR); SUBTY=B003; BEGDA= P0298-BEGDA) визначити код з поля HRP1001-SOBID для HRP1001-SCLAS=A

За допомогою ФМ RH\_READ\_INFITY (PLVAR=01; OTYPE=A; OBJID= HRP1001-SOBID (для HRP1001-SCLAS=A); INTY=1640; BEGDA= P0298-BEGDA мінус 1 день) визначити та вивести відповідне значення для коду з поля P1640-LBCOND:

P1640-LBCOND=0 вивести «НУ»;

P1640-LBCOND=1 вивести «ШУ»;

P1640-LBCOND=2 вивести «ВУ»;

P1640-LBCOND=3 вивести «ОШУ»;

P1640-LBCOND=4 вивести «ОВУ»;

[A48] Інформація щодо суб'єкта декларування на старій посаді. Для P0001-PLANS (ФМ HR\_READ\_INFOTYPE на дату BEGDA, ENDDA = P0298-BEGDA мінус 1 день) в таблиці HRP1010-OBJID перевіряємо значення для полів. В разі, якщо HRP1010-SUBTY=9004, HRP1010-HILFM=01, то виводимо текст: «Є суб'єктом декларування»; в іншому випадку виводити: «Не є суб'єктом декларування».

[A49] Інформація щодо суб'єкта декларування на новій посаді. Для P0001-PLANS (ФМ HR\_READ\_INFOTYPE) в таблиці HRP1010-OBJID перевіряємо значення для полів. В разі, якщо HRP1010-SUBTY=9004, HRP1010-HILFM=01, то виводимо текст: «Є суб'єктом декларування»; в іншому випадку виводити: «Не є суб'єктом декларування».

### 3.2 Вибір інструментарію розробки

Як показали проведені дослідження, SAP HANA - це платформа для зберігання, обробки даних та створення застосунків, що включає в себе не тільки базу даних в пам'яті, але й інструменти розробки користувацьких

застосунків (мови програмування, фреймворк UI5, середовище розробки, різноманітні послуги - попередня аналітика, графи, текстовий аналіз, обробка потоків даних, тощо) [19].

SAP Cloud Platform - це, по суті, SAP HANA, розгорнута в хмарі SAP або партнерів. SAP Cloud Platform включає в себе набір сервісів, доступних розробнику:

- сервіси БД (HANA, Sybase ASE, MongoDB, Redis, Postgres);
- сервіси розробки та запуску програм на «своїх» мовах (Java, Javascript) та за моделлю BYOL (на базі Cloud Foundry);
- додаткові сервіси: попередня аналітика, сервіс аутентифікації, розробка мобільних застосунків, бізнес-аналітика, гейміфікація, рахунки, IoT, інтеграція тощо (близько 40 сервісів, що встановлюються SAP, а також близько 1000 сервісів, розроблених партнерами).

SCP має низку властивих хмарі переваг - немає «заліза», вже все налаштовано, просте адміністрування під конкретні завдань, оплата за фактом використання і т.д. В даний час набір сервісів SCP дещо ширше, ніж набір сервісів в SAP HANA, встановленої на обладнанні клієнта, наприклад, сервіси гейміфікації, білінгу є тільки в хмарній версії. Також в SCP є багато сервісів, реалізованих як готовий до використання застосунок. Наприклад Business Object Cloud (хмарна аналітика), Cloud Platform Integration (інтеграційна шина), BUILD (швидке прототипування призначених для користувача інтерфейсів), Smart Business Services (моніторинг ключових показників) і т.д [19].

ABAP є ключовою мовою розробки бізнес-застосунків SAP, в тому числі застосунків нового покоління, таких як S/4 HANA. Однак SAP розширює свій портфель можливостей, включаючи інші мови до складу базових засобів розробки. Цим SAP знижує поріг входу для сторонніх розробників, які мають досвід створення унікальних бізнес-рішень і бажають використовувати встановлену інфраструктуру SAP в якості бізнес-сервісів, але не ставлять собі за мету вивчення ABAP.

Cloud Platform пропонує можливості для зберігання та обробки даних у SAP HANA та ASE, а також відкриті джерела систем PostgreSQL, MongoDB та Hadoop. SCP підходить для розробки мобільних та HTML-застосунків, які можуть бути легко інтегровані з будь-якими хмарними або локальними системами, завдяки вбудованому в інтеграцію сервісу SCP. На даний момент SCP пропонує близько 40 різноманітних сервісів, включаючи IoT [19], алгоритми прогнозування, машинний переклад та інші методи машинного навчання, має власну вебсторінку розроблених WebIDE та інструмент для швидкого прототипування Build, підтримує розробку застосунків на Java, XSJS, C ++, Python, Ruby, Node.js, Go, PHP, .Net [20].

ABAP/4 (Advanced Business Application Programming) - пропрієтарна внутрішня мова програмування високого рівня німецької софтверної компанії SAP. Поряд з Java є мовою створення застосунків для SAP NetWeaver Application Server. Синтаксис найбільш близький до мови програмування COBOL. Мова реалізує роботу з внутрішніми структурами даних, інтерфейсом користувача SAP ERP, транзакціями, звітами, інтерфейсами завантаження і вивантаження даних. Використовується виключно для бізнес-застосунків і проміжного ПЗ компанії SAP. Має можливості для об'єктно-орієнтованого програмування. Оригінальний текст ABAP компілюється (генерується) в виконуваний ABAP байт-код, відомий як «report load». Виконується на віртуальній машині [21]- [23].

### **3.3 Методика розробки хмарного застосунку**

Існуючі методики розробки застосунків, що представлено в [24]- [26], не враховують можливість розробки бізнес застосунків з використанням мови ABAP/4, тому мають бути удосконалені.

Cloud Foundry (CF) є глобальним стандартом розробки хмарних застосунків, який дозволяє розробникам приділяти більше уваги поставленим

завданням і полегшує процес розгортання і управління застосунками. Використовувані для розробки сервіси і застосунки наступні.

Застосунок Product List - використовується в якості прикладу, що містить в собі список персоналу. Він буде працювати в середовищі Cloud Foundry хмарної платформи SAP, з використанням різних служб, таких як PostgreSQL для зберігання в хмарі, Application Logging (логування застосунку) та інші. Це застосунок типу SpringBoot з простим призначенням для користувача інтерфейсом.

Сервіс Application Logging дозволяє створювати, переглядати та аналізувати логи застосунків. Вона заснована на платформі логування з відкритим вихідним кодом Elasticsearch, Logstash, Kibana (Elastic Stack).

Сервіс Application Autoscaler використовується для автоматичного масштабування застосунку і для масштабування пов'язаних застосунків, заснованих на користувальницьких політиках. Динамічне масштабування екземпляра застосунку гарантує відсутність проблем забезпечення продуктивності при збільшенні навантаження і не вийде з ладу. Динамічне масштабування також дає гарантію того, що застосунок використовує оптимальні ресурси.

Методика розробки бізнес-застосунку містить наступні основні етапи (рис.3.1).

Етап 1. Підготовка середовища розробки. Після авторизації в системі, необхідно на домашній сторінці обрати Cloud Foundry Trial (рис.3.2). Після вибору регіону, надається акаунт, субакаунт, організація та простір в обраному регіоні (рис.3.3).

Після цього треба перейти до створеного простору, натиснувши Go to Space. Для налаштування локального середовища розробки потрібно встановити на ПК Cloud Foundry Command Line Interface (CF CLI) та середовище AВАР 4.

Етап 2. Клонування шаблону застосунку.

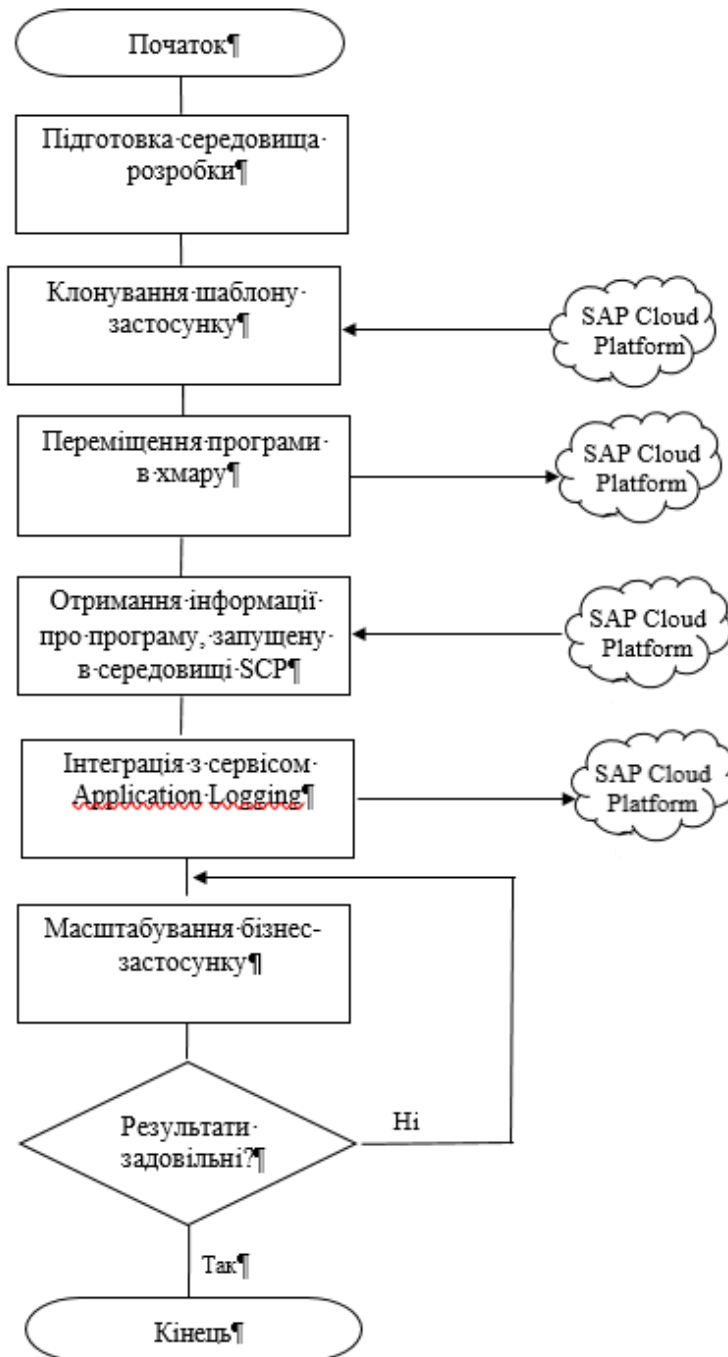


Рисунок 3.1 – Схема алгоритму методики розробки бізнес-застосунку

Далі виконується клонування шаблону застосунку Product List (рис.3.4), його імпортування в робоче середовище (File -> Import...), модифікація на основі моделі BYOL та подальше складання (рис.3.5). Після успішного складання, застосунок треба завантажити в пробний обліковий запис в середовищі Cloud Foundry платформи SCP. Це можна зробити за допомогою інтерфейсу командного рядка Cloud Foundry (CF CLI).

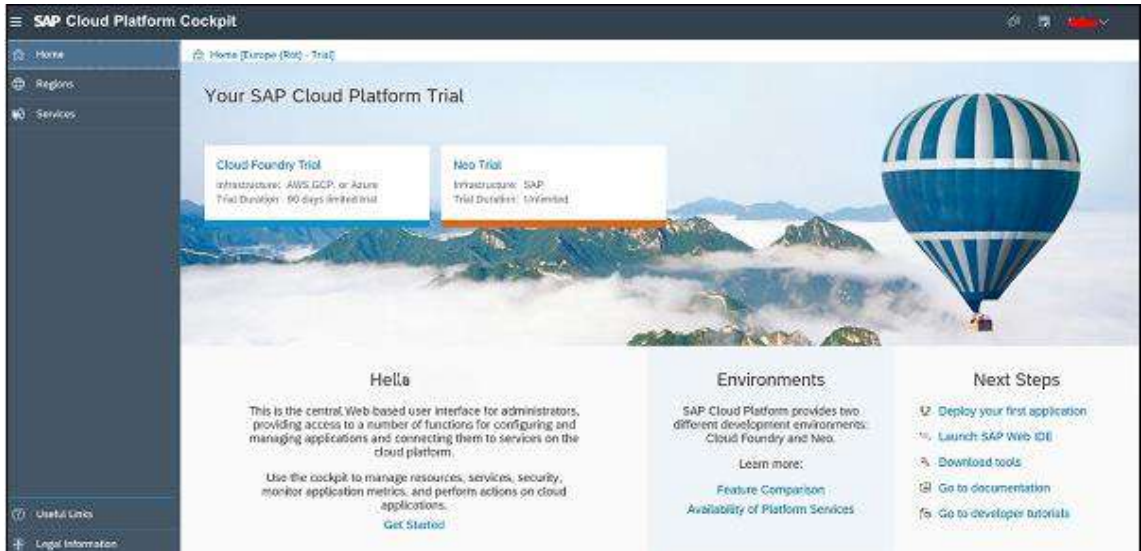


Рисунок 3.2 – Стартова сторінка SAP Cloud Platform

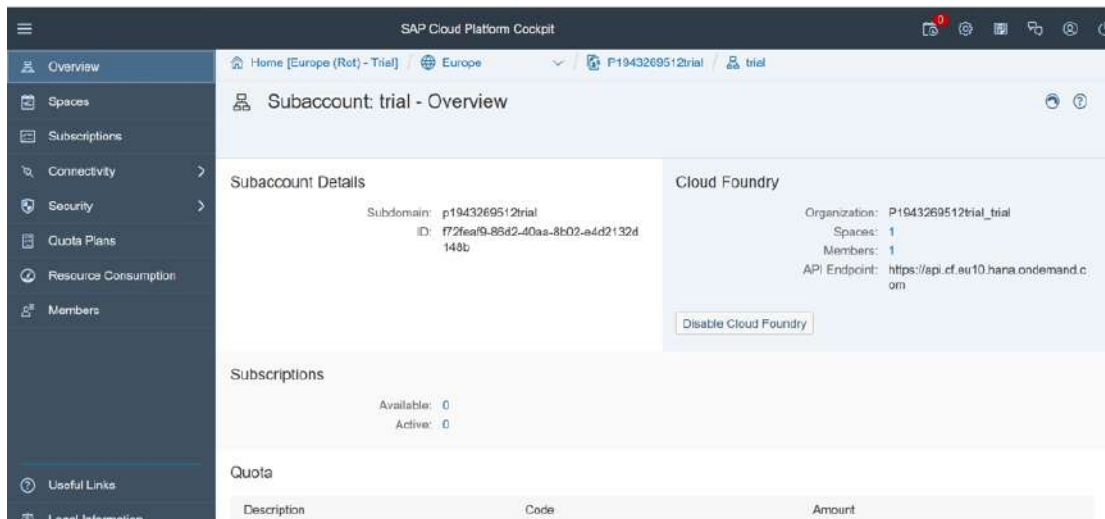


Рисунок 3.3 – Панель управління SCP

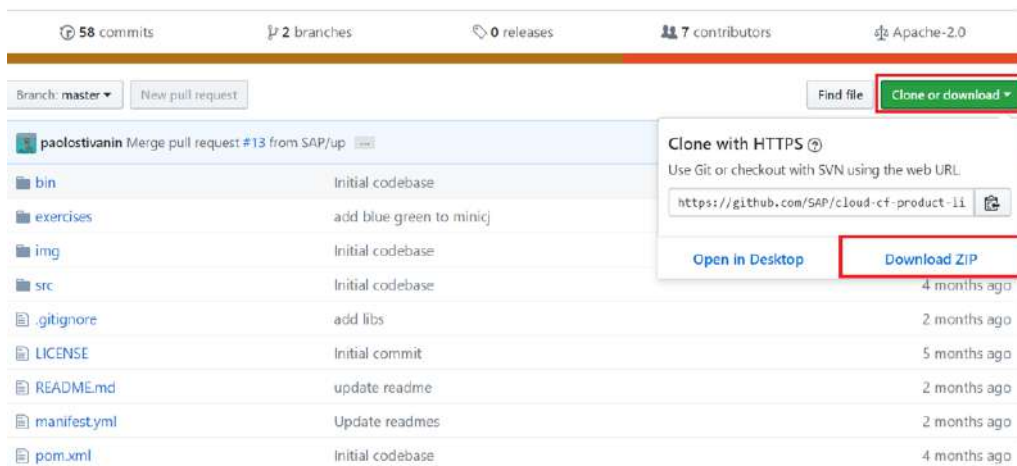


Рисунок 3.4 – Клонування шаблону застосунку

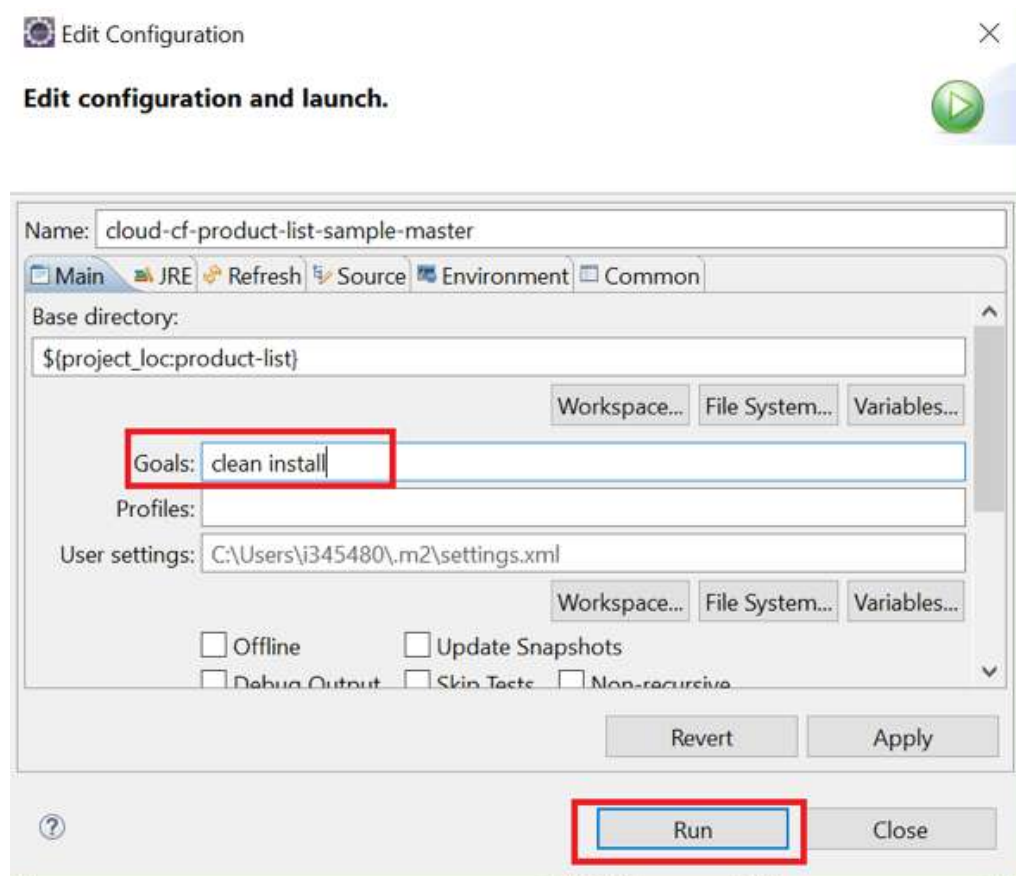


Рисунок 3.5 – Складання проєкту

Етап 3. Переміщення програми в хмару. Перед завантаженням застосунку в середовище Cloud Foundry потрібно прописати параметри програми в командному рядку для команди `push`, що дозволяє запуснути застосунок в хмарі, або використовувати файл дескриптора застосунка (`manifest.yml`), встановивши в ньому параметри для переміщення, що дозволить не вводити їх заново кожен раз перед використанням команди `push`. Тепер застосунок запущено в середовищі Cloud Foundry платформи SAP Cloud Platform (рис.3.6).

Етап 4. Отримання інформації про програму, запущену в середовищі SAP Cloud Platform. Додаткову інформацію про застосунки і служби, запущені в пробному обліковому записі в середовищі Cloud Foundry, а також основні метрики і логи (споживання процесора, пам'яті і диска) можна дізнатися як за допомогою CF CLI, так і через панель управління SAP Cloud Platform.

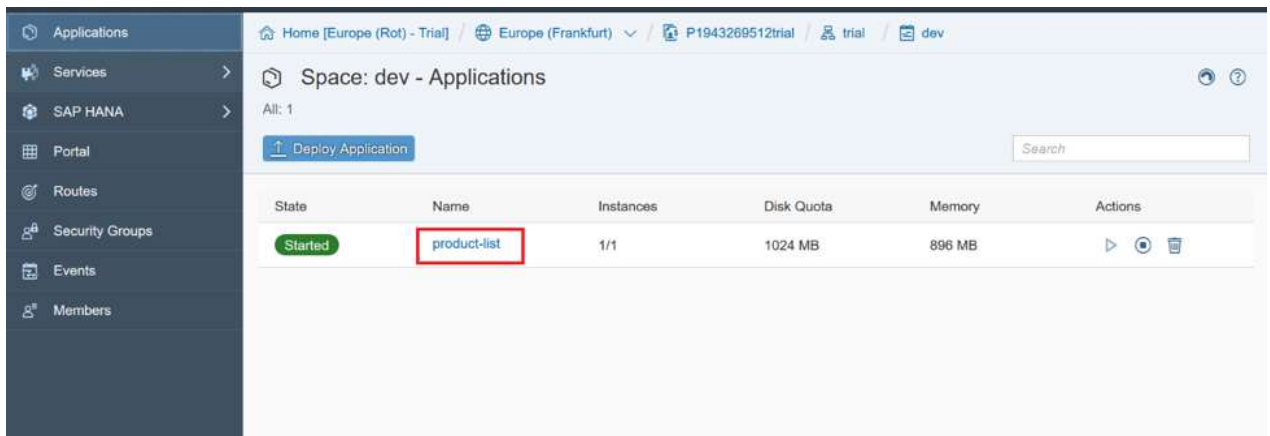


Рисунок 3.6 – Робочий простір з переміщенням застосунком

Перевірка працездатності програми - це процес моніторингу, який постійно перевіряє статус застосунка, запущеного в Cloud Foundry. У середовищі Cloud Foundry таймаут за замовчуванням становить 60 секунд, а максимальний таймаут - 180 секунд. У Cloud Foundry доступні три типи перевірки працездатності:

- http - перевірка працездатності типу http виконується за допомогою запиту GET на налаштовану кінцеву точку http; коли перевірка працездатності отримує відповідь HTTP 200, застосунок вважається «здоровим»; конфігурована кінцева точка повинна відповідати протягом 1 секунди, щоб вважатися здоровою;

- port - це перевірка за замовчуванням; цей тип перевірки здійснює TCP-з'єднання з портом або портами, налаштованими для застосунка (за замовчуванням 8080); TCP-з'єднання повинно бути встановлено за 1 секунду, щоб застосунок вважався «здоровим»;

- process - Diego гарантує, що будь-який процес, оголошений для застосунка, буде працювати; якщо процес завершується, Diego зупиняється і видаляє екземпляр застосунку.

Тип перевірки працездатності може бути вказаний при переміщенні застосунку в хмару з допомогою команди push:

```
cf push APP_NAME -u HEALTH-CHECK-TYPE -t HEALTH-CHECK-TIMEOUT
```

Етап 5. Інтеграція з сервісом Application Logging (логування застосунку). Для цього у вихідному коді застосунку має підтримуватися бібліотека SAP - підтримка логуювання для Cloud Foundry. Для генерації логів достатньо викликати застосунок через URL-адресу в панелі управління SAP Cloud Platform (рис. 3.7).

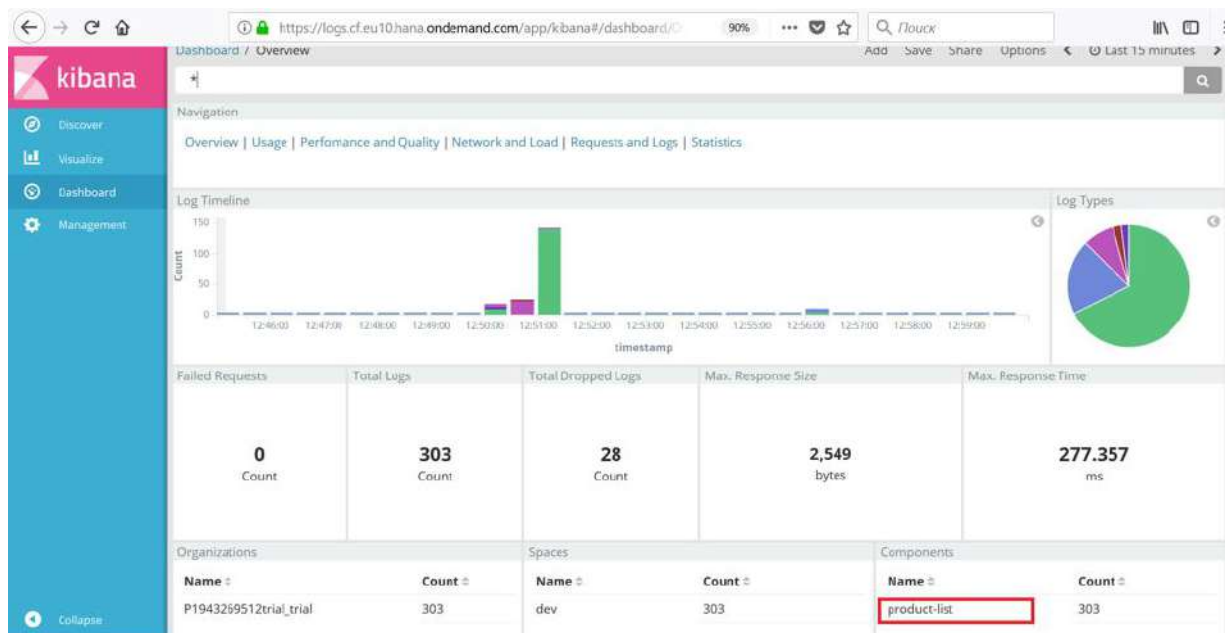


Рисунок 3.7 – Статистика застосунку

Етап 6. Масштабування бізнес-застосунку. Навантаження системи, кількість та характер завдань - всі ці фактори можуть змінювати простір і пам'ять на диску, що використовуються застосунком. Для багатьох застосунків збільшення доступного простору, пам'яті або запуск більшої кількості примірників може підвищити загальну продуктивність. Аналогічно, запуск додаткових примірників програми може дозволити застосунку регулювати збільшення навантаження та одночасних запитів. Це регулювання називається масштабуванням застосунку. Можливо масштабувати застосунок таким чином, щоб він був пристосований до змін у трафіку, вручну через CF CLI або через панель управління SAP Cloud Platform. Крім того, можна налаштувати автоматичне використання сервісу Application Autoscaler (автоматичне масштабування застосунку) платформи

SCP. Зокрема, вертикальне масштабування застосунка змінює обмеження на дисковий простір або обмеження пам'яті, яке Cloud Foundry застосовується до всіх екземплярів програми (рис.3.8 - 3.9)

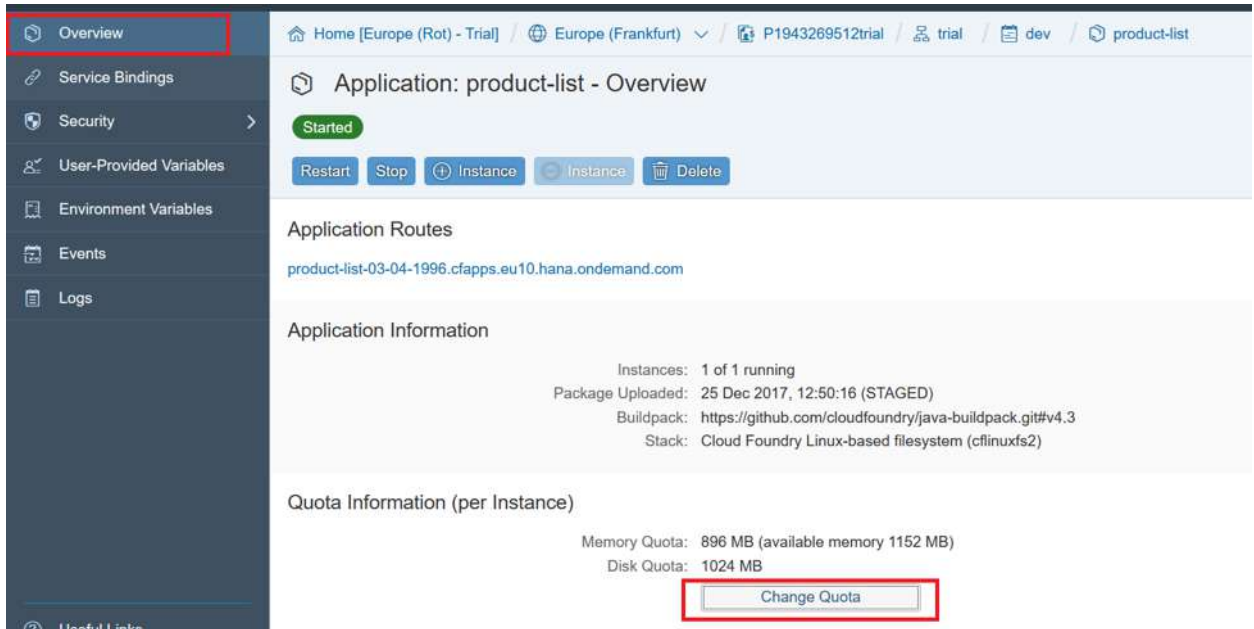


Рисунок 3.8 – Масштабування застосунку

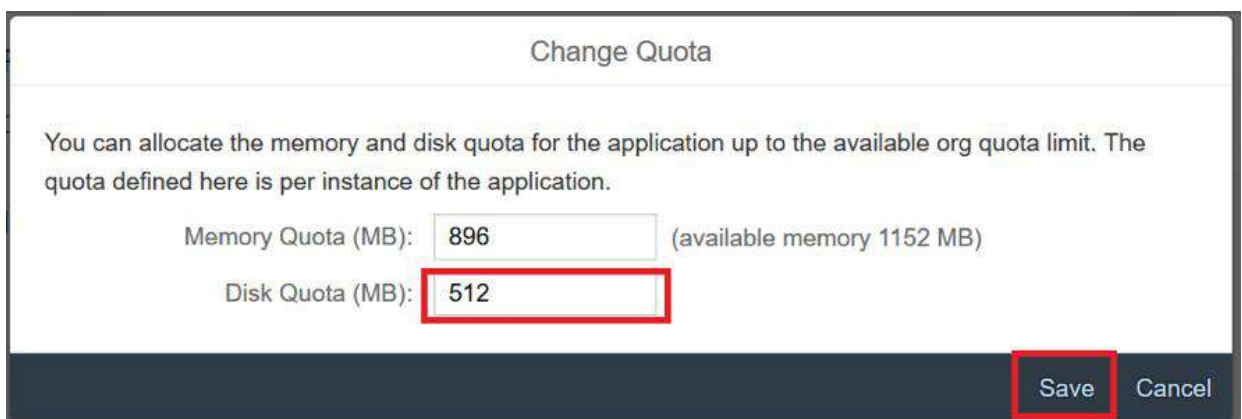


Рисунок 3.9 – Зміна квоти пам'яті на диску

Горизонтальне масштабування застосунка створює або видаляє екземпляри застосунка. Вхідні запити застосунка автоматично регулюються по навантаженню у всіх примірниках застосунка, і кожен екземпляр обробляє завдання паралельно з іншими екземплярами. Додавання примірників дозволяє застосунку обробляти більшу кількість трафіку і запитів. Існує два

способи вертикального масштабування програми: за допомогою CF CLI або через панель управління в SAP Cloud Platform.

Недоліком ручного способу масштабування застосунка є те, що ці дії виконуються тільки в тому випадку, якщо розробник помічає, що продуктивність застосунку погіршилася. Існує ризик, що застосунок вийде з ладу, якщо масштабування не буде виконано у відповідний час. Тому іноді краще мати можливість автоматичного масштабування застосунку за допомогою служби Application Autoscaler платформи SCP.

Етап 7. Перевірка результатів. Після перевірки відповідності застосунку вимогам, у випадку задовільних результатів, процес розробки завершується, інакше, потрібно повернутися на етап масштабування для внесення коректив.

Отже, середовище Cloud Foundry на базі платформи SCP є відмінним рішенням для розробки хмарного бізнес-застосунку й додати в нього можливості моніторингу, масштабування та оновлення.

### **3.4 Особливості програмної реалізації**

Узагальнену структурну схему програми наведено на рисунку 3.10. Основними складовими застосунку є:

- модуль формування персонального наказу на переведення;
- модуль формування групового наказу на переведення;
- модуль виводу на друк.

Робота застосунку здійснюється в хмарному сервісі SAP Cloud Platform.

Вхідними даними програми є дані, що наведені в таблиці 3.2.

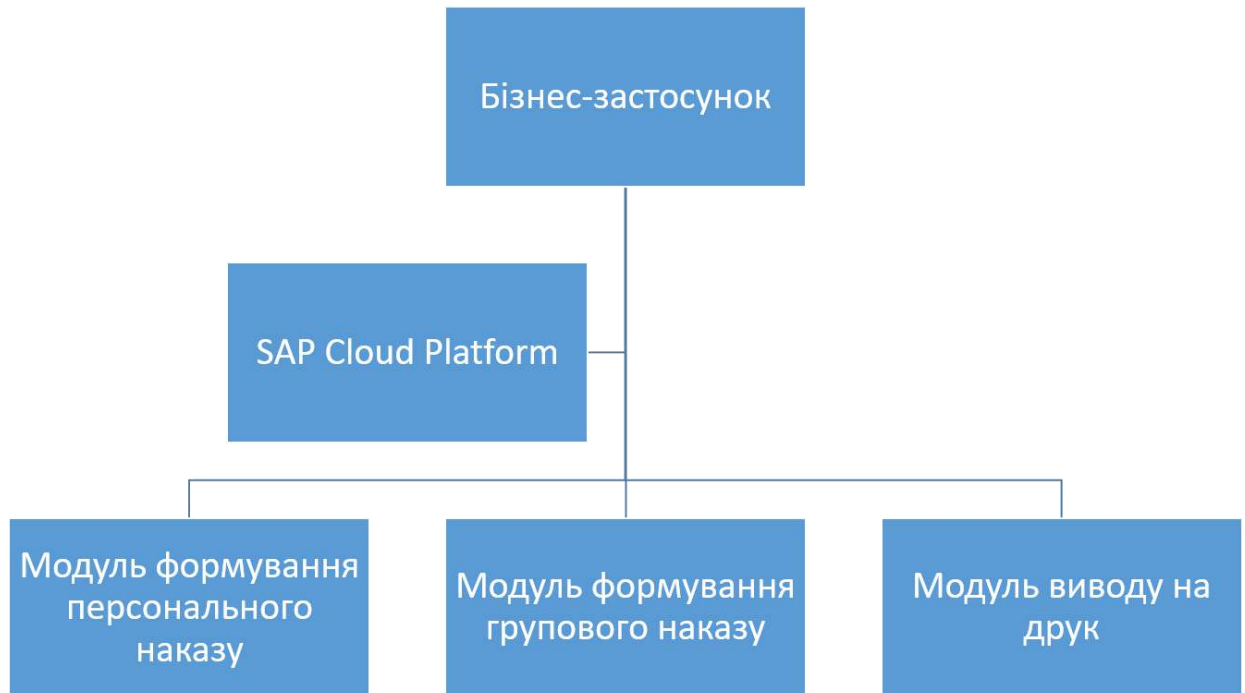


Рисунок 3.10 – Узагальнена структура програми

Таблиця 3.2 - Вхідні дані

Поле	Назва	Тип	Засіб пошуку	Примітка
1	2	3	4	5
PNPBEGDA	Дата початку	BEGDA	Стандартне	Автоматично заповнюється датою початку заходу
PNPENDDA	Дата завершення	ENDDA	Стандартне	Автоматично заповнюється датою початку заходу
PNPPERNR	Табельний номер	PERNR_D	Стандартне	Автоматично заповнюється табельним номером вибраного співробітника
MASSN	Вид заходу	MASSN	Стандартне	Автоматично заповнюється видом заходу

Продовження табл. 3.2

1	2	3	4	5
MASSG	Причина заходу	MASSG	Стандартне	Автоматично заповнюється причиною заходу
ORDNU	Номер наказу	p33_ordnu	Стандартне	Автоматично заповнюється номером наказу
ORDDT	Дата наказу	p33_orddt	Стандартне	Автоматично заповнюється датою наказу

Дані, що використовуються P0298, P0002, P0001, P0007, P0298, P0016, P0007, P0008, P9005, HRP1010, HRP1640. Для роботи створеної програми формування та друку наказів ZBHR\_PRINT\_03OR. необхідно виконати налаштування згідно з таблицею 3.3.

Текст програми наведено в Додатку А. Основні екранні форми бізнес-застосунку наведено на рисунках 3.11.-3.13. Результати роботи програми наведено в Додатку Б.

Таблиця 3.3 - Зв'язок заходу та програми друку

Назва заходу	Параметр
Групування країн	36
Код країни	UA
Вид заходу	02
Причина заходу	01, 02
Завершення строку дії	31.12.2020
Початок строку дії	01.09.2020
Ім'я АВАР-програми	ZBHR_PRINT_03OR
Тип формуляру	Інший

Основні дані керування персоналом | Редагувати | Перейти до | Додатково | Утиліти | Налаштування | Система | Довідка

Відобразити основні дані HR

Табельний номер: 1000017 | Труд.дог.: 01000017 30000101 01000017

Таб: 1000017 | Ім'я: Іванов Іван Іванович | 0101 | Промислово-виро...

Гру: 1 | Штатні пр... | Орг.о: 40000095 | Щось інше | 0100 | ВП «Запорізька АЕ...

Під: 11 | Керівник... | Штат.: 30000101 | Фахівець тест1 | К-ть: 2 розряду

Основні дані | Додаткові дані | Дані обліку часу | Розрахунок ЗП

Текст інфо-типу	С..
Заходи	✓ ^
Організаційне присвоювання	✓ v
Особисті дані	✓
Адреси	✓
<b>Накази щодо персоналу</b>	✓
Компоненти контракту	✓
Документи і сертифікати	✓
Освіта	✓ ^
Трудова книжка	✓ v

Період

Період: 3 | 01.01.1800 | До: 31.12.9999

Сьогодні | Поточний тижд.

**Усе** | Поточний місяць

3 поточної дати | Ост.тижд.

До поточ.дати | Останній місяць

Поточний період | Поточний рік

Обрати

Прямий вибір

Інфотип: Накази щодо персоналу | ТЗП:

Рисунок 3.11 – Форма основних даних щодо адміністрування персоналу

Основні дані керування персоналом | Редагувати | Перейти до | Середовище | Система | Довідка

Накази щодо персоналу Огляд

Вибрати

Таб: 1000017 | Ім'я: Іванов Іван Іванович | 0101 | Промислово-виро...

Гру: 1 | Штатні пр... | Орг.о: 40000095 | Щось інше | 0100 | ВП «Запорізька АЕ...

Під: 11 | Керівник... | Штат.: 30000101 | Фахівець тест1 | К-ть: 2 розряду

Обрати | 01.01.1800 | до: 31.12.9999 | Тип:

Дата зап.	С..	Тип заходу	П..	Причина для заходу	Пі...	Назва	Номер наказу
23.10.2020	30	Відпустка	09	СоцВідДляДоглядЗад...		Стандартне зам...	134В
<b>24.07.2020</b>	<b>02</b>	<b>Переведення</b>	<b>02</b>	<b>Тимчасово</b>		Стандартне зам...	130-КВ
18.06.2020	01	Прийм на ро...	01	Безстроковий трудов...		Стандартне зам...	121-КВ

Рисунок 3.12 – Форма перегляду наказів щодо персоналу

Основні дані керування персоналом | Редагувати | Перейти до | Середовище | Система | Довідка

Накази щодо персоналу Вибір.

**Друквати документ**

Знайти за

- Особа
  - Комплексний засіб пошуку
  - Шуканий термін
  - Вільний пошук

Таб. 1000017 | Ім'я: Іванов Іван Іванович | 0101 Промислово-виро...  
 Гру. 1 | Штатні пр... Орг.о 40000095 | Щось інше | 0100 ВП «Запорізька АЕ...  
 Під. 11 | Керівник... Штат. 30000101 | Фахівець тест1 | К-ть 2 розряду

Почат. 24.07.2020 | Д.зм. 06.10.2020 | 100207

**Накази щодо персоналу**

Тип операції	02	Переведення
Причина операції	02	Тимчасово
Подробиці причини	04	Тимчасово

Тимчасово

Номер наказу: 130-КВ  Груповий ордер  
 Дата наказу: 24.07.2020  Підписано  
 Місце зберігання наказу:  Підписано

Дата ознайомлення:

Спец.нотатки:

**Керівник**

Табельний номер	1005584
Ім'я	Сніцин Микола Володимирович
Штатна пос.кер.	НЗЦАЕСС

**Стандартні тексти**

Підстава для наказу:

Зміст змін в наказі:

**Керівники**

Керівник 2	2
Керівник 3	11
Керівник 4	0
Керівник 5	0
Відповідальний	18

Рисунок 3.13 – Форма відображення наказу та виведення на друк

### 3.5 Висновки до розділу 3

В даному розділі відповідно до мети та задач магістерської роботи виконано удосконалення методики розробки бізнес-застосунку на основі хмарного сервісу. Розроблено структуру застосунку, визначено вхідні та вихідні дані, виконано програмну реалізацію на основі клону. Перевірку працездатності застосунку та його масштабування виконано з використанням сервісу Cloud Foundry та служби Application Autoscaler платформи SCP.

## 4 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА

В результаті виконання дипломної кваліфікаційної роботи магістра розроблено програмний продукт, що відноситься до функціонального напрямку «Адміністрування персоналу» та призначений для автоматичного формування документу «Наказ про переведення» за заданим шаблоном для окремого працівника або для групи працівників.

Програма має на меті підвищити ефективність процесу адміністрування персоналу за рахунок скорочення часу на формування документів.

### 4.1 Планування розробки програмного продукту

Весь комплекс розробки програмного продукту можна розділити на етапи. Для кожного етапу вказується трудомісткість, кількість виконавців і тривалість робіт [27]. В розробці беруть участь програміст протягом 1 місяця і аналітик протягом 0,25 місяця. Розробка ПЗ починається з 1 жовтня і має бути виконана до 1 листопада 2020 року.

Тривалість робіт визначають за формулою (4.1):

$$T_{\text{ц}} = \frac{Q}{R}, \quad (4.1)$$

де  $T_{\text{ц}}$  – тривалість циклу, днів;

$Q$  – трудомісткість, людино-днів;

$R$  – кількість виконавців, чел.

Отриману інформацію зведено у таблицю 4.1.

Таблиця 4.1 – Характеристика робіт з розробки програми

Найменування робіт	Трудомісткість		Виконавці	Тривалість днів
	Люд.- дні	%		
1. Аналіз предметної області	10	37,04	Програміст Аналітик	5
2. Визначення вимог до програмного продукту	2	7,41	Програміст	2
3. Проектування структури програми	3	11,11	Програміст	3
4. Розробка схеми функціонування програми	2	7,41	Програміст	2
5. Написання програмного коду	4	14,81	Програміст	4
6. Тестування і налагодження програмного продукту	3	11,11	Програміст	3
7. Складання програмної документації	3	11,11	Програміст	3
Всього	27	100		22

За даними таблиці 4.1 складається зведений стрічковий графік планування розробки програмного продукту, який представляє собою таблицю, в першому стовпці якої розміщені в порядку збільшення термінів початку виконання всі види робіт, а навпроти - календарний період їх виконання [27]. Даний графік наведено в таблиці 4.2.

Таблиця 4.2 – Зведений стрічковий графік планування розробки програмного продукту

Найменування робіт	Календарний період, дні				
	30.09-04.10	07.10-11.10	14.10-18.10	21.10-25.10	28.10-01.11
1. Аналіз предметної області	■ ■				
2. Визначення вимог до програмного продукту		■			
3. Проектування структури програми		■			
4. Розробка схеми функціонування програми			■		
5. Написання програмного коду			■	■	
6. Тестування налагодження програмного продукту				■	
7. Складання програмної документації					■

#### 4.2 Визначення витрат на розробку програми

Для визначення витрат на розробку програми складається калькуляція кошторисної вартості робіт, яка включає наступні статті:

- основна заробітна плата;
- додаткова заробітна плата;
- відрахування на соціальне страхування і в інші фонди;
- витрати на спеціальне обладнання;

- матеріали і комплектуючі вироби;
- накладні витрати;
- податки [27].

#### 4.2.1 Розрахунок основної заробітної плати

Витрати за цією статтею складаються з планового фонду зарплати всіх категорій працівників, зайнятих в розробці програми [27]. Розрахунок зарплати ведеться на підставі даних про трудомісткості, представлених в таблиці 4.3.

Таблиця 4.3 – Розрахунок основної заробітної плати

Посада виконавця	Чисельність, люд.	Місячний оклад	Кількість місяців роботи	Сума ЗП, грн..
Програміст	1	5000	1	5000
Аналітик	1	9000	0,25	2250
Всього	2			7250

#### 4.2.2 Розрахунок додаткової заробітної плати

Додаткову заробітну плату приймають рівною 10% від основної заробітної плати працівників і розраховують за формулою (4.2):

$$ЗП_{\text{дод}} = ЗП_{\text{осн}} \cdot 0,1 \quad (4.2)$$

Підставивши величину основної заробітної плати в дану формулу, отримуємо:

$$ЗП_{\text{дод}} = 7250 \cdot 0,1 = 725 \text{ грн.}$$

### 4.2.3 Відрахування на єдиний соціальний внесок

Ці відрахування визначають як 22% від сум основної та додаткової зарплати з урахуванням премій і доплат, за формулою (4.3):

$$ОТ = (ЗП_{осн} + ЗП_{дод}) \cdot 0.22 \quad (4.3)$$

Після підстановки отримаємо:

$$ОТ = (7250 + 725) \cdot 0.22 = 1754,5 \text{ грн.}$$

### 4.2.4 Визначення витрат на матеріали

Використовуються 2 найменування матеріалів: usb-флеш – накопичувач – 400 грн.; папір 90 грн (1 упаковка).

Витрати на матеріали розраховують за формулою (4.4):

$$M = \sum_{i=1}^n (C_i \cdot N_i \cdot (1 + K_{Т.З.}) - C_{i0} \cdot N_{i0}), \quad (4.4)$$

де  $M$  – витрати на матеріали, покупні напівфабрикати і комплектуючі вироби, грн.;

$K_{Т.З.}$  – коефіцієнт, що враховує транспортно-заготівельні витрати;

$C_i$  – ціна  $i$ -го найменування матеріалу, напівфабрикату і комплектуючого,

грн.;

$N_i$  – потреба в  $i$ -му матеріалі, напівфабрикаті і комплектуючому;

$C_{i0}$  – ціна зворотних відходів  $i$ -го найменування матеріалу, грн;

$N_{i0}$  – кількість зворотних відходів  $i$ -го найменування;

$n$  – кількість найменувань матеріалів, напівфабрикатів і комплектуючих.

Виконаємо відповідні обчислення:

$$Ц_{i0} = 0; N_{i0} = 0; K_{т.з.} = 0,05;$$

$$M = (1 + 0.05) \cdot (400 + 90) = 514,5 \text{ грн.}$$

Всього, витрати на матеріали становлять 514,5 грн.

#### 4.2.5 Витрати на спеціальне обладнання

У цю статтю входять витрати на придбання, транспортування, монтаж і налагодження нестандартного обладнання. Практично, в даному випадку, в цій статті враховуються витрати на оплату машинного часу електронно-обчислювальної машини (ЕОМ) для написання і налагодження даної програми. Для чого необхідно скласти кошторис «Витрат на утримання та експлуатацію обладнання» виходячи з якої визначиться вартість одного машино-години роботи ПК, після множення якої на машинний час пішло на написання і налагодження програми отримаємо витрати на оплату машинного часу [27].

Амортизаційні відрахування визначають за формулою (4.5):

$$A = \Phi_6 \cdot N_a, \quad (4.5)$$

де  $\Phi_6$  – балансова вартість обчислювальної техніки, грн.;

$N_a$  – норма амортизаційних відрахувань на повне відновлення обчислювальної техніки.

Балансова вартість обчислювальної техніки становить 10000 грн.

Отримуємо після підстановки:

$$A = 1000 \cdot 0,25 = 2500 \text{ грн.}$$

Статтю «Експлуатація обладнання» розраховують підсумовуванням витрат на електроенергію і допоміжні матеріали за формулою (4.6):

$$C_e = N_n \cdot \Phi_{\text{еф}} \cdot K_{\text{зв}} \cdot K_{\text{зм}} \cdot C_e, \quad (4.6)$$

де  $N_n$  – номінальна потужність ЕОМ, кВт;

$\Phi_{\text{еф}}$  – річний ефективний фонд часу роботи ЕОМ, машино-годину;

$K_{\text{зв}}$  – середній коефіцієнт завантаження за часом;

$K_{\text{зм}}$  – коефіцієнт завантаження по потужності;

$C_e$  – ціна одного кВт \* год електроенергії, грн. / (кВт \* год).

Номінальна потужність ЕОМ - 0,3 кВт. Річний ефективний фонд часу роботи ЕОМ становить 1800 годин. Середні коефіцієнти завантаження за часом і за потужністю рівні відповідно 0,9 і 0,6. Ціна однієї кіловат-години електроенергії складає 2,11 грн. Отримуємо після підстановки:

$$C_e = 0,4 \cdot 1800 \cdot 0,9 \cdot 0,6 \cdot 2,11 = 615,27 \text{ грн.}$$

Витрати за статтею "Заробітна плата обслуговуючих робітників та відрахування на соціальне страхування в інші фонди" визначають за формулою (4.7):

$$ЗП_{\text{обс}} = \Phi ЗП_p \cdot (1 + K_{\text{відр}}) \cdot \frac{t_{\text{обсл}}}{\Phi_{\text{еф.обс}}}, \quad (4.7)$$

де  $\Phi ЗП_p$  – річний фонд заробітної плати (основної і додаткової) обслуговуючих робітників, грн.;

$K_{\text{відр}}$  – коефіцієнт, що враховує відрахування на єдиний соціальний внесок;

$t_{\text{обсл.}}$  – час протягом року, необхідний на технічне обслуговування ЕОМ, год/рік;

$\Phi_{\text{еф.обсл.}}$  – річний ефективний фонд часу обслуговуючого персоналу, год/рік.

Місячна заробітна плата обслуговуючого персоналу становить 4173 грн., а річний фонд заробітної плати відповідно дорівнює 50076 грн. Річний ефективний фонд робочого часу обслуговуючого ПК працівника дорівнює 1750 год / рік на обслуговування одного ПК витрачається по 2 годині на місяць, що в рік становить 24 годин. Отримуємо:

$$ЗП_{\text{обсл.}} = 50076 \cdot (1 + 0,22) \cdot \frac{24}{1750} = 837,84 \text{ грн.}$$

Таблиця 4.4 – Кошторис витрат на утримання і експлуатацію обладнання

Найменування статей витрат	Сума, грн.
Амортизація обладнання	2500
Експлуатація обладнання / крім витрат на поточний ремонт /	615,27
Заробітна плата основна і додаткова обслуговуючих робітників з відрахуваннями на соціальне страхування і в інші фонди	837,84
Поточний ремонт обладнання	300
Інші витрати	191,7
Всього	4444,81

Стаття «Поточний ремонт обладнання» приймається рівною 3% від балансової вартості обладнання і складає 300 грн. Стаття «Інші витрати» приймається рівною 5% від суми всіх попередніх статей витрат на утримання і експлуатацію обладнання. Сума всіх попередніх статей дорівнює 3834,19

грн., 5% від суми складають 191,7 грн. Розраховані статті витрат на утримання і експлуатацію обладнання внесені в таблицю 4.4.

Витрати на оплату машинного часу ЕОМ для написання та налагодження програми розраховується за формулою (4.8):

$$C_{\text{мо}} = P_{\text{екс}} \cdot t_{\text{мо}}, \quad (4.8)$$

де  $C_{\text{мо}}$  – витрати на оплату машинного часу, грн.;

$P_{\text{екс}}$  – експлуатаційні витрати на одну годину машинного часу цієї цифрової ЕОМ, грн. / машино-год.;

$t_{\text{мо}}$  – машинний час цифрової ЕОМ для написання і налагодження даного програмного продукту, машино-год.

Експлуатаційні витрати на одну годину машинного часу використовуваної ЕОМ розраховують діленням суми витрат за кошторисом "Витрати на утримання та експлуатацію обладнання" з табл. 4.4 на річний ефективний фонд часу роботи ЕОМ.

$$P_{\text{екс}} = \frac{4444,81}{1800} = 2,46 \text{ грн./машино-год}$$

ЕОМ експлуатується 22 дні в одну зміну, що складає в сумі 176 годин. Таким чином, витрати на оплату машинного часу складають:

$$C_{\text{мо}} = 2,46 \cdot 176 = 432,96 \text{ грн.}$$

#### 4.2.6 Розрахунок накладних витрат

До накладних витрат відносяться витрати на загальне управління і загальногосподарські потреби (заробітна плата апарату управління, канцелярські витрати і т.д.), утримання та експлуатацію будівель [27].

Накладні витрати включаються до вартості розробки програми непрямым шляхом - у відсотках до основної заробітної плати розробників. В даному випадку накладні витрати становлять 40% до основної заробітної плати розробників, що складає 2900 грн.

Результати визначення витрат на розробку програми у вигляді калькуляції кошторисної вартості робіт наведені в таблиці 4.5.

Таблиця 4.5 – Калькуляція кошторисної вартості робіт по розробці програми

Найменування статей витрат	Сума, грн..	%
Основна заробітна плата	7250	52,69
Додаткова заробітна плата	725	5,26
Відрахування на соцстрах та ін..	1754,5	12,75
Матеріали та комплектуючі	514,5	3,73
Витрати на спец. обладнання	432,96	4,47
Накладні витрати	2900	21
Всього	13576,96	100

### 4.3 Ціна ПЗ

Плановий прибуток розраховується за формулою (4.9):

$$P_{\text{п}} = C_{\text{п}} \cdot Y_{\text{рп}}, \quad (4.9)$$

де  $P_{\text{п}}$  – прибуток від реалізації ПЗ, грн.;

$C_{\text{п}}$  – собівартість ПЗ (сума витрат на реалізацію), грн.;

$Y_{\text{рп}}$  – рівень рентабельності ПЗ, приймається 30 %.

За формулою (4.9) визначаємо прибуток від реалізації ПЗ:

$$\Pi_{\pi} = 13576,96 \cdot 0,3 = 4073,08 \text{ грн.}$$

До ціни ПЗ входить податок на додаткову вартість (ПДВ), який розраховується за формулою (4.10):

$$\text{ПДВ} = D_{\text{в}} \cdot N_{\text{пдв}}, \quad (4.10)$$

де  $N_{\text{пдв}}$  – податок на додаткову вартість, грн;

$D_{\text{в}}$  – додаткова вартість, грн.;

Норматив ПДВ приймається рівним 20%.

Додаткова вартість розраховується за формулою (4.11):

$$D_{\text{в}} = C_{\pi} + \Pi_{\text{р}} \quad (4.11)$$

Отримаємо після підстановки:

$$D_{\text{в}} = 13576,96 + 4073,08 = 17650,04 \text{ грн.}$$

За формулою (4.10) визначимо ПДВ:

$$\text{ПДВ} = 17650,04 \cdot 0,2 = 3530,00 \text{ грн.}$$

Ціна ПЗ являє собою суму собівартості, планованого прибутку та податку на додаткову вартість і складає 21180,04 грн.

#### 4.4 Розрахунок економічної ефективності програмного продукту

Програма дає можливість скоротити час формування документу «Наказ на переведення» (робота виконується за 15 годин в місяць замість 50).

Економічна ефективність програми досягається за рахунок автоматизації процесу формування документів. Економія часу з 50 до 15 годин в місяць, що в сумі приводить до економії 2625 грн в місяць протягом 12 місяців у рік (якщо враховувати що зарплата становить 75 грн./год ). Річна економія заробітної плати с урахуванням нарахувань визначається за формулою (4.12):

$$E_{зп} = E_{зп.міс.} \cdot (1 + K_{відр.}) \cdot n, \quad (4.12)$$

де  $E_{зп}$  – річна економія по заробітній платі з відрахуваннями на соціальні заходи, грн.;

$E_{зп.міс.}$  – місячна економія заробітної плати при впровадженні програми, грн.

В результаті отримуємо:

$$E_{зп} = (50 - 15) \cdot 75 \cdot 1,22 \cdot 12 = 38430 \text{ грн.}$$

Річна економія визначається за формулою (4.13):

$$E = (T_{M_1} - T_{M_2}) \cdot P_{екс.ктз} + E_n^a \cdot \frac{\Phi_{б.ктз} \cdot (T_{M_1} - T_{M_2})}{\Phi_{эф.ктз}} - \frac{S_{рп}}{T_c} + E_{зп} \quad (4.13)$$

де  $T_{M_1}$ ,  $T_{M_2}$  – машинний час, необхідний для рішення поставлених завдань, відповідно в старому та у новому варіантах, машино-год./рік;

$P_{\text{екс.кТЗ}}$  – експлуатаційні витрати, що доводяться на 1 год машинного часу ЕОМ, грн/ машино-год.;

$E_n^a$  – нормативний коефіцієнт ефективності капітальних вкладень у засоби автоматизації;

$\Phi_{\text{б.кТЗ}}$  – балансова вартість ЕОМ, грн.;

$\Phi_{\text{еф.кТЗ}}$  – річний ефективний фонд часу роботи ЕОМ, машино-год.;

$S_{\text{рп}}$  – сумарні витрати на розробку програми, грн.;

$T_c$  – термін служби впроваджуваної програми до її морального зношування, років;

Після підстановки отримаємо:

$$E = (50-15) \cdot 12 \cdot 2,46 + 0,15 \cdot 10000 \cdot ((50-15) \cdot 12 / 1800) - (21180,04)/5 + 38430 = 35577,19 \text{ грн.}$$

Коефіцієнт ефективності вкладених коштів в програму  $E$  і термін їх окупності  $T_{\text{ок}}$  розраховують по формулам:

$$E = \frac{E}{S_{\text{рп}}} \quad (4.14)$$

$$T_{\text{ок}} = \frac{S_{\text{рп}}}{E} \quad (4.15)$$

Таким чином отримаємо:

$$E = 35577,19 / 21180,04 = 1,67,$$

$$T_{\text{ок}} = 21180,04 / 35577,19 = 0,59 \text{ років.}$$

Техніко-економічні показники наведено в таблиці 4.6.

Таблиця 4.6 – Техніко-економічні показники

Показники	Одиниці виміру	Варіанти	
		базовий	новий
1. Витрати на розробку програми	грн.		21180,04
2. Експлуатаційні витрати	грн.	40406	4828,81
3. Річна економія	грн.		35577,19
4. Термін окупності	років		0,59
5. Тривалість розрахунків		більше	менше
6. Достовірність інформації		менше	більше

#### 4.5 Висновки до розділу 4

Термін окупності програми складає 0,59 року. Коефіцієнт ефективності капітальних вкладень становить 1,67, що більше нормативного (0,15). Річна економія при використанні програми становить 35577,19 грн. Таким чином, впровадження програмного продукту є економічно доцільним.

## 5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

### 5.1 Аналіз потенційних небезпек

У ході виконання магістерської роботи досліджується та практично використовується хмарна платформа SAP для розробки програмного забезпечення.

При роботі в офісному приміщенні на розробника постійно або періодично діють наступні небезпечні і шкідливі чинники.

Підвищені статичні навантаження користувачів ПК, які пов'язані з тривалим перебуванням в однотипній позі та повторних однотипних рухів кистями рук. Порухення правил ергономіки може призвести до ураження рухового апарату та тунельного синдрому.

Незадовільні ергономічні характеристики робочого місця внаслідок нераціонального планування робочого місця, що може призвести до механічних травм, уражень електричним струмом та порушень кістково-м'язового апарату.

Небезпеки, пов'язані електростатичним випромінюванням, що може призвести до загальних захворювань.

Можливість враження електричним струмом внаслідок порушень правил електробезпеки або несправності електроприладів, що може призвести до електричних травм або летального випадку.

Незадовільні параметри мікроклімату в приміщенні внаслідок не ефективної роботи апаратури для опалювання та повітрообміну, що може призвести до загальних захворювань

Перевищення рівнів вібрації застарілого обладнання, що може призвести до сильної втоми, незначних змін функцій організму (порушення серцевої діяльності, нервової системи).

Недостатнє освітлення робочої зони внаслідок несправності

освітлювальних приладів, що може призвести до погіршення зору

Можливість загорянь внаслідок порушень правил пожежної безпеки, що може призвести до пожеж.

Неправильні дії персоналу під час надзвичайних ситуацій (НС) внаслідок відсутності управління та паніки, що може призвести до травм та летальних наслідків.

## **5. 2 Заходи із забезпечення безпеки**

Забезпечення ергономічності робочого місця виконано згідно з [28], де зазначена заборона захаращеності робочого місця та дотримання нормативів щодо площини робочого місця на одного працівника. Також вимоги до організації робочого місця зазначені в [29].

Оснащення офісного приміщення виконано раціонально з точки зору ергономіки. Зручне та раціональне розташування робочих місць визначається відстань між ними. Згідно з вимогами, щодо виробничих приміщень площа одного робочого місця  $\geq 6 \text{ м}^2$ , а об'єм  $\geq 20 \text{ м}^3$ .

Конструкція робочого столу відповідає сучасним вимогам ергономіки і забезпечує оптимальне розміщення на робочій поверхні використовуваного обладнання (дисплеїв, клавіатури, принтерів) і документів.

Висота робочої поверхні робочого столу регулюється в межах 680...800 мм, а ширина і глибина забезпечує можливість виконання операцій у зоні досяжності моторного поля (рекомендовані розміри: 600...1400 мм, глибина - 800...1000 мм).

Робочий стіл має простір для ніг заввишки не менше ніж 600 мм, завширшки не менше ніж 500 мм, завглибшки (на рівні колін) не менше ніж 450 мм, на рівні простягнутої ноги - ніж 650 мм.

Робочий стілець працівника відповідно до [28] є підйомно-поворотним, з можливістю регулювання висоти та куту нахилу сидіння та спинки. Регулювання за кожним із параметрів здійснюється незалежно, легко

і надійно фіксуватися. Шаг регулювання елементів стільця становить: для лінійних розмірів - 15...20 мм, для кутових - 2...5<sup>0</sup>.

Висота спинки стільця становить  $(300 \pm 20)$  мм, ширина - не менше ніж 380 мм, радіус кривизни горизонтальної площини - 400 мм. Кут нахилу спинки регулюється в межах 1...30<sup>0</sup> від вертикального положення. Відстань від спинки до переднього краю сидіння регулюється в межах 260...400 мм [28].

За ступенем небезпеки ураження електричним струмом, відповідно до правил улаштування електроустановок (ПУЕ) [30], приміщення обчислювального центру (ОЦ) відноситься до категорії приміщень без підвищеної небезпеки.

Заходи по забезпеченню електробезпеки розроблено у відповідності до [31]. За способом захисту людини від ураження електричним струмом – відеотермінали, ЕОМ, периферійні пристрої ЕОМ та устаткування для обслуговування, ремонту та налагодження ЕОМ відповідають першому класу захисту та заземлені відповідно до [32].

Для забезпечення електробезпеки у ОЦ використано наступні технічні заходи та засоби електробезпеки:

- недоступність до струмопровідних частин;
- захисне відключення;
- ізоляція струмопровідних частин;
- захисне заземлення;
- блокування, занулення за ПУЕ;
- захисне розділення електричних мереж [33];
- компенсація ємкісних струмів замикання на землю;
- вирівнювання потенціалів тощо.

Електромережа у приміщенні офісу вмикається/вимикається за допомогою пускової апаратури.

У приміщенні передбачено використання тільки трьохполюсних розеток, один з полюсів яких підключено до шини захисного заземлення. З

цим полюсом електрично з'єднані металеві частини обчислювальних машин, які можуть виявитися під напругою.

Кожний блок живлення комп'ютера або периферійного пристрою має сітьовий фільтр. Конденсатори цього фільтру призначені для шунтування високочастотних поміх живильної мережі на землю через дріт «земля» трьохполюсної вилки та розетки. Цей дріт може бути з'єднаний з заземлюючим контуром і з «нулем» силової мережі. При відсутності гальванічного зв'язку заземлюючого дроту з шиною захисного заземлення, на корпусі електронного пристрою може з'явитися напруга порядку 110 В, внаслідок поділення напруги мережі 220 В конденсаторами сітьового фільтру, що працює у даному випадку як ємнісний роздільник напруги [34].

Відповідно до ПУЕ опір захисного заземлення у приміщенні ОЦ не перевищує 4 Ом [30].

Електрична ізоляція забезпечується конструкціями пристроїв і способами їхнього підключення. Відповідно до ПУЕ в електроустановках до 1000 В мінімальне значення опору ізоляції силових і освітлювальних електропроводок на ділянці між суміжними запобіжниками або за останніми запобіжниками між будь-яким проведенням і землею становить не менш 0,5 МОм [30].

Накопичена статична електрика, зокрема, на екрані монітора притягує пил, бруд та інші частини присутні в повітрі. Причому електризується не тільки екран, а і повітря на робочому місці, а також одяг користувача, особливо якщо він з синтетичного та шерстяного матеріалу.

В якості заходів захисту запропоновано:

- кілька разів на протязі робочого дня мити руки і обличчя водою, а після закінчення роботи вимити руки й лице з милом;
- щоденно протирати екран монітора, клавіатуру, пристрій «миша» та, якщо є, приекранний фільтр антистатичною серветкою;
- щоденно в приміщенні з ПК проводити вологе прибирання;
- установити нейтралізатори статичної електрики;

- підтримувати у приміщенні вологість повітря зазначену в нормативних документах;

- користувачу ПК бажано носити одяг з природних (льняних) волокон.

Для запобігання статистичного навантаження при користуванні ПК рекомендовано використовувати перерви в роботі 10 хв. через кожні дві години. Синдром зап'ястного каналу, або тунельний синдром зап'ястя, який може бути наслідком хронічної травми, трапляється у людей внаслідок тривалої роботи з мишею: постійні напруга і здавлювання приводить до мікротравм, здавлювання нерву прилеглими оточуючими тканинами, через що виникає набряк.

Для запобігання тунельного синдрому потрібно в першу чергу правильно підібрати робоче місце та встановити комп'ютер. Потрібно мати спеціальний стіл та стілець, рекомендується мати можливість регулювати висоту робочої поверхні від 655 до 740 мм. Важливим є простір для ніг користувача: висота – не менше 600мм, ширина – не менше 600 мм, глибина на рівні коліна – не менше 400 мм, на рівні підлоги – не менше 600 мм. Також рекомендується мати на комп'ютерному робочому місці спеціальний стілець, який: зможе змінювати свою висоту, має можливість регулювати уклін опори для спини (від 0 до 200) , може повертатися, має опір для ліктів. Не погано також мати підставку для ніг. Висоту стільця треба підбирати таким чином, щоб забезпечити перпендикулярність лінії зору користувача екрану монітора.

Щоб тунельний синдром не турбував, потрібно дотримуватися кількох правил організації робочого місця:

- оптимальна висота клавіатури від підлоги – 65-75 см;

- наявність ергономічних і зручних особисто для вас миші і клавіатури;

- можливість регулювання висоти і нахилу клавіатури (відстань від поверхні стола до середини клавіатури – не більше 30 мм, кут підйому клавіатури – від 2 ° до 15 °);

- наявність у клавіатури підставки для рук;
- наявність килимка для миші з захистом від тунельного синдрому (спеціальний виступ забезпечує правильне положення кисті);
- наявність стільця або крісла з підлокітниками;

При роботі з мишкою і клавіатурою також слід дотримуватися певних правил. Коли ви набираєте текст, рука повинна бути зігнута в лікті під прямим кутом ( 90 град), а при роботі з мишкою стежте, щоб кисть була прямою і лежала на столі якнайдалі від краю. До речі, час роботи з комп'ютером слід обмежити до дійсно необхідного.

Щоб попередити тунельний синдром потрібно робити спеціальні вправи для кистей. Комплекс вправ потрібно повторювати приблизно кожні 45 хвилин, тривалість однієї вправи складає 1-2 хв [34].

### **5.3 Заходи з виробничої санітарії та гігієни праці**

У офісному приміщенні, згідно [35] передбачене природне та штучне освітлення. Штучне освітлення в приміщенні, здійснено системою загального рівномірного освітлення. Значення освітленості на поверхні робочого столу в зоні розміщення документів становить 200-300 лк. Як джерела штучного освітлення в приміщенні застосовані люмінесцентні лампи типу ЛБ. При застосуванні яких дотримано наступних умов:

- температура навколишнього повітря не нижче, ніж 5°C;
- напруга на освітлювальних приладах не менше, ніж 90% від номінальної.

Розрахуємо систему загального рівномірного штучного освітлення в офісному приміщенні.

Приймаємо розміри приміщення довжина  $A = 7$  м, ширина  $B = 5$  м, висота  $H = 4$  м. Висота робочої поверхні складає  $h = 0,7$  м. Коефіцієнти відбиття поверхонь приміщення:  $\rho_c=50\%$ ;  $\rho_{cm}=30\%$ ;  $\rho_n=10\%$ . Розряд зорової роботи –III в.

Згідно з [35] рівень нормованого загального освітлення в приміщенні становить  $E_n = 200$  лк.

Обираємо світильник типу ЛВП, призначений для приміщень висотою до 4,5 м. IP=54 – ступінь захисту світильника від пилу та бризок води,  $[L/h] = 1,3$  – числове значення коефіцієнта світильника.

Коефіцієнт запасу  $k_3$  враховує зниження рівня освітленості з часом в результаті забруднення та старіння ламп, світильників і поверхонь приміщення, приймається в залежності від виробничих умов. Приймаємо  $k_3 = 1,4$ .

Коефіцієнт нерівномірності (мінімального) освітлення  $z$  (відношення середньої освітленості до мінімальної освітленості), для люмінесцентних ламп низького тиску, як правило дорівнює  $z = 1,1$ .

Розрахуємо кількість рядів світильників у приміщенні за формулою (5.1):

$$N_{\text{екс}} = \frac{B}{(H - h_p) * (L/h)} \quad (5.1)$$

Після підстановки отримаємо:

$$N_{\text{екс}} = \frac{5}{(4 - 0,7) * 1,3} = 2 \text{ ряди}$$

Визначимо максимально допустиму відстань між рядами світильників за формулою (5.2):

$$L_{\text{max}} = \frac{B}{N_p}, \quad (5.2)$$

де  $B$  – ширина приміщення,

$N_p$  – кількість рядів.

Виконаємо розрахунок:

$$L_{\max} = \frac{5}{2_p} = 2,5 \text{ м}$$

Розраховуємо висоту підвісу світильника над робочою поверхнею за формулою (5.3):

$$h = \frac{L_{\max}}{[L/h]} \quad (5.3)$$

Після підстановки отримаємо:

$$h = \frac{2,5}{1,3} = 1,92 \text{ м}$$

Знайдемо висоту звисання світильника від стелі за (5.4):

$$h_3 = H - h_p - h \quad (5.4)$$

де  $H$  – висота приміщення,

$h_p$  – висота робочої поверхні,

$h$  – висота підвісу світильника над робочою поверхнею.

Проведемо обчислення:

$$h_3 = 4 - 0,7 - 1,92 = 1,38 \text{ м}$$

Чисельне значення індексу приміщення визначаємо за рівнянням (5.5):

$$i = \frac{AB}{h(A+B)}, \quad (5.5)$$

де  $A$  – довжина приміщення,  $m$ ;

$B$  – ширина приміщення,  $m$ ;

$h$  – висота розміщення світильників над робочою поверхнею,  $m$ .

Виконаємо розрахунок:

$$i = \frac{7 \cdot 5}{1,92 \cdot (7 + 5)} = 1,52$$

Значення коефіцієнта використання світлового потоку  $\eta$  вибирається в залежності від виду джерела світла, типу обраного світильника, коефіцієнтів відбиття поверхонь приміщення ( $\rho_c$ ,  $\rho_{ст}$ ,  $\rho_{п}$ ) та індексу приміщення. В нашому випадку  $\eta = 50\%$ .

Визначаємо сумарний світловий потік освітлювальної установки у даному приміщенні згідно з (5.6):

$$\Phi_{\Sigma} = \frac{E_n \cdot S \cdot k_3 \cdot z}{\eta}, \quad (5.6)$$

де  $\Phi_{\Sigma}$  – розрахункове значення сумарного світлового потоку,  $лм$ ;

$E_n$  – нормоване значення освітленості,  $лк$ ;

$S$  – площа освітлюваної поверхні,  $m^2$ ;

$k_3$  – коефіцієнт запасу;

$z$  – коефіцієнт нерівномірності (мінімальної) освітленості;

$\eta$  – коефіцієнт використання світлового потоку.

Після підстановки даних отримаємо:

$$\Phi_{\Sigma} = \frac{200 \cdot 7 \cdot 5 \cdot 1,4 \cdot 1,1}{0,50} = 21560 \text{ лм}$$

Визначаємо умовну загальну кількість світильників у приміщенні за рівнянням (5.7):

$$N^* = AB/L^2_{max} \quad (5.7)$$

Проведемо розрахунок:

$$N^* = \frac{7 \cdot 5}{2,5 \cdot 2,5} = 5,6 \approx 6 \text{ шт.}$$

Оскільки воно кратне числу рядів  $N_p = 2$ .

Розрахуємо світловий потік умовного джерела світла за формулою (5.8):

$$\Phi_{л}^{\otimes} = \frac{\Phi_{\Sigma}}{N_{л}}, \quad (5.8)$$

де  $\Phi_{\Sigma}$  – розрахункове значення сумарного світлового потоку, *лм*;

$N_{л}$  – загальна кількість ламп у приміщенні, *шт*;

Загальна кількість ламп у приміщенні розраховується згідно з (5.9):

$$N_{л} = N^* \cdot n, \quad (5.9)$$

де  $n$  – кількість ламп у світильнику (в ЛВП світильнику знаходиться 2 лампи).

Після підстановки визначимо:

$$N_{л} = 6 \cdot 2 = 12 \text{ шт.}$$

Тоді, за формулою (5.8) отримаємо:

$$\Phi_{\text{л}} = \frac{21560}{12} = 1797 \text{ лм}$$

Обираємо тип стандартної лампи з найближчим значенням фактичного світлового потоку лампи  $\Phi_{\text{л}}$  (наприклад, лампа ЛБ із  $P_{\text{л}}=30$  Вт,  $\Phi_{\text{л}}=1880$  лм). Знайдемо коефіцієнт  $m$  (співвідношення між розрахунковим світловим потоком лампи  $\Phi_{\text{л}}^{\otimes}$  та фактичним світловим потоком вибраної стандартної лампи  $\Phi_{\text{л}}$ ) за формулою (5.10):

$$m = \frac{\Phi_{\text{л}}^{\otimes}}{\Phi_{\text{л}}} \quad (5.10)$$

Після підстановки отримаємо:

$$m = \frac{1797}{1880} = 0,96$$

Визначаємо оптимальну (фактичну) кількість світильників у приміщенні за формулою (5.11):

$$N = N^* m \quad (5.11)$$

Підставимо дані та отримаємо:

$$N = 6 * 0,96 \approx 6 \text{ шт.}$$

Фактична кількість ламп у приміщенні розраховується за (5.12):

$$N_{\text{фл}} = N_{\text{ф}} \cdot n \quad (5.12)$$

Виконаємо розрахунок:

$$N_{\text{фл}} = 6 \cdot 2 = 12 \text{ шт.}$$

Для загального рівномірного освітлення офісного приміщення використовуємо 6 світильників типу ЛВП, розташованих по 3 у кожному ряді.

Метеорологічні умови в виробничих приміщеннях - температура повітря, відносна вологість повітря й швидкість його переміщення відповідають вимогам [36].

Роботи в офісному приміщенні, належать до категорії Іб - легка робота, тому передбачені наступні оптимальні значення параметрів мікроклімату:

- у холодну пору року: температура 22-24° С; відносна вологість: 40-60%; швидкість переміщення повітря: 0,1 м/с

- у теплу пору року: температура 23-25° С; відносна вологість: 40-60%; швидкість переміщення повітря: 0,2 м/с.

Рівні вібрації під час виконання робіт з ЕОМ у виробничих приміщеннях не перевищують допустимих значень, визначених у [37].

Нормування технологічної вібрації (загальної і локальної) відбувається:

- в залежності від її спрямування у кожній октавній смузі (1,6-1000 Гц);

- з середньоквадратичними віброшвидкостями  $(1,4-0,28) \cdot 10^{-2}$  м/с; логарифмічними рівнями віброшвидкості (115-109 дБ);

- віброприскоренням  $(85-0,1)$  м/с<sup>2</sup>.

## **5.4 Заходи безпеки у надзвичайних ситуаціях**

### **5.4.1 Заходи з пожежної безпеки**

Клас пожежі для офісних приміщень встановлюється згідно [38] – пожежі твердих речовин, переважно органічного походження, горіння яких супроводжується тлінням (деревина, пластмаси, папір) і визначається як клас А.

Категорія офісного приміщення за вибухопожежною небезпекою визначається згідно [39], як пожежонебезпечність адміністративних приміщень.

Офісне приміщення обладнано автоматичними пожежними сповіщувачами, що реагують на підвищення температури, дим, полум'я, наприклад, сповіщувачі різних моделей.

Для відводу надлишкового тепла від ЕОМ задіяні кондиціонування повітря і система вентиляції, що обладнана пристроєм, який забезпечує автоматичне відключення вентиляції, в разі виникнення пожежі.

Напруга до електроустановок подається по кабельним мережам, що прокладені під технологічною з'ємною підлогою з рівнем вогнестійкості не менше 0,5 год.

До первинних засобів тушіння пожеж, призначених для ліквідації невеликих загорянь використовуються пожежні стволи, внутрішні пожежні водопроводи, вогнегасники. Згідно [38] при захисті від пожежі приміщення з ПЕОМ використовуються вуглекислотні вогнегасники ВВК-1,4, ВВК-2 або аерозольні водопінні вогнегасники ВВПА-400.

Згідно з НАПБ Б.03.001-2004 приміщення з ЕОМ необхідно оснащувати переносними вуглекислотними вогнегасниками з розрахунку один вогнегасник ВВК-1,4 або ВВК-2 або ВВПА-400 на три ПЕОМ, але не менше ніж один вогнегасник, перелічених типів на приміщення.

У випадку пожежі передбачено шляхи евакуації робітників проходи, проїзди, евакуаційні виходи. Евакуаційні виходи розташовано розосереджено

у кількості не менше двох на споруду.

Коридори та проходи, що призначені для евакуації, мають як можна меншу довжину та мінімальну кількість поворотів. На усій протяжності проходу відсутні пороги або проміжні ступені.

Важливу роль у забезпеченні безпечного виходу людей грає протидимовий захист евакуаційних ходів. У будинках висотою до 9 поверхів незадимлюваність сходових кліток на час евакуації досягається їхньою ізоляцією від підвалів, горищ і поверхів. Для цього влаштовані відособлені входи в підвали, вхід на сходову клітку з поверхів здійснюють через тамбур-шлюз із підпором повітря, відокремлюють горища від сходових кліток перекриттями з негорючих матеріалів [37].

#### **5.4.2 Заходи з цивільного захисту**

Вплив електромагнітного імпульсу на елементи виробництва.

Ядерні вибухи в атмосфері й більш високих шарах призводять до виникнення потужних електромагнітних полів з довжиною хвиль від 1 до 1000 м і більше. Ці поля через короткочасне існування називають електромагнітним імпульсом (ЕМІ). ЕМІ виникає при ядерному вибуху у воєнний час, у мирний час — при випробуванні ядерної зброї або ядерних аваріях і катастрофах в атмосфері й космосі.

Основною причиною виникнення ЕМІ тривалістю менше 1 с вважають взаємодію гамма-променів і нейтронів ядерного вибуху з атомами газів повітря, внаслідок чого з них вибиваються електрони (ефект Комптона) і хаотично розлітаються в середовищі позитивно заряджених атомів газів. Важливе значення має також виникнення асиметрії в розподілі просторових електричних зарядів, пов'язаних з особливостями поширення гамма-променів і утворення електронів.

При наземному і низькому повітряному вибуху вражаюча дія ЕМІ спостерігається на відстані кількох кілометрів від центру вибуху. Під час

ядерного вибуху на висотах від 3 до 25 км утворюється симетричне джерело генерації, але радіус поширення ЕМІ залишається обмеженим внаслідок сильного поглинання гамма-випромінювання в щільних шарах атмосфери.

Уражаюча дія ЕМІ обумовлена виникненням напруги і струмів у провідниках різної довжини, розміщених у повітрі, землі.

Уражаюча дія ЕМІ в приземній області й на землі пов'язана з акумулюванням його енергії довгими металевими предметами, рамними і каркасними конструкціями, антенами, лініями електропередачі та зв'язку, в них виникають сильні наведені струми, які руйнують підключене електронне та інше чутливе устаткування. У районі дії ЕМІ безпосередній контакт людини зі струмопровідними предметами небезпечний.

ЕМІ уражає радіоелектронну і радіотехнічну апаратуру. В провідниках індукуються високі напруги і струми, які можуть призвести до постійних або тимчасових пошкоджень ізоляції кабелів, відключення реле і переривників, пошкодження елементів зв'язку, магнітних запам'ятовуючих пристроїв у ЕОМ і системах передачі даних тощо. Найбільш уразливими елементами обладнання є напівпровідникові прилади — транзистори, діоди, кремневі випрямлячі, інтегруючі ланцюги, цифрові процесори, управляючі й контрольні прилади. Чутливі до пошкодження ЕМІ транзистори звукової частоти, перемикаючі транзистори, інтегруючі ланцюги та ін.

Найбільш стійкі до ЕМІ вакуумні електронні прилади, які виходять із ладу при енергії 1 Дж. Величина енергії ЕМІ залежить від ширини періоду частот антенних систем.

Більшість систем зв'язку працюють у діапазоні частот від середніх до ультрависоких і будуть пошкодженими залежно від робочого періоду частот. Радіолокаційні системи менше пошкоджуються від ЕМІ, тому що вони працюють у періоді частот, де щільність енергії ЕМІ невелика. Іскріння, яке виникає під впливом високого електричного поля ЕМІ, може спричинити спалахування парів бензину та інших налив у сховищах.

Якщо ядерний вибух стався поблизу лінії електропостачання, зв'язку

великої довжини, то наведені в них напруги можуть поширюватися по проводах на багато кілометрів, пошкоджувати апаратуру й уражати людей, які знаходяться на безпечній відстані відносно інших уражаючих факторів ядерного вибуху.

ЕМІ небезпечний і за наявності міцних споруд, розрахованих на стійкість проти ударної хвилі наземного ядерного вибуху, проведеного на відстані кількох сотень метрів [40].

### **5.5 Висновки до розділу 5**

В даному розділі виконано розробку заходів з забезпечення безпеки, виробничої санітарії та гігієни праці при роботі в офісі. Крім того, запропоновано заходи безпеки в надзвичайних ситуаціях: з пожежної безпеки та цивільного захисту.

## ВИСНОВКИ

В результаті проведених досліджень показано актуальність впровадження хмарних технологій для удосконалення ІТ-ландшафту інтелектуального підприємства, зокрема підвищення рівня автоматизації документообігу. При цьому, моделі PaaS і IaaS можуть використовуватись для розробки та експлуатації різноманітних бізнес-застосунків. Для підвищення ефективності процесів розробки і експлуатації ПЗ рекомендується використовувати набір практик DevOps та фреймворк CALMS, який дозволяє оцінити готовність переходу до DevOps та може застосовуватися як модель для виявлення можливих удосконалень. SAP Cloud Platform відноситься до хмарних платформ, що забезпечують середовище виконання для розробки, тестування та управління програмними застосунками. Важливою особливістю SCP є підтримка моделі BYOL, що забезпечує використання будь-якої мови програмування, наприклад, PHP, Python та інші. Оскільки ABAP/4 є ключовою мовою розробки бізнес-застосунків SAP, було прийнято рішення про розробку хмарного бізнес-застосунку для адміністрування персоналу на його основі. Перевірка працездатності застосунку та його масштабування виконано з використанням сервісу Cloud Foundry та служби Application Autoscaler платформи SCP.

Наукове значення роботи полягає в тому, що удосконалено методику розробки бізнес-застосунку на основі хмарного сервісу, що відрізняється від існуючих підтримкою мови програмування ABAP/4 на основі моделі BYOL.

Практичне значення роботи полягає в тому, що розроблений на основі удосконаленої методики бізнес-застосунок дозволяє підвищити рівень автоматизації документообігу підприємства за рахунок зменшення витрат часу та помилок при формуванні наказів на переведення персоналу та внесенні змін у суміжні документи.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Как с помощью SAP Cloud Platform создать расширение для цифрового ядра [Электрон. ресурс]. – Режим доступа: <https://habr.com/ru/company/sap/blog/502572/>.
2. Облачная пирамида: IAAS, PAAS и SAAS [Электрон. ресурс]. – Режим доступа: <https://gigacloud.ua/ru/blog/navchannja/hmarna-piramida-iaas-paas-i-saas>.
3. Модели услуг IBM Cloud: IaaS, PaaS и SaaS [Электрон. ресурс]. – Режим доступа: <https://www.ibm.com/ru-ru/cloud/learn/iaas-paas-saas>.
4. SAP представил новые IoT-сервисы на конференции SAP Leonardo Live [Электрон. ресурс]. – Режим доступа: <https://news.sap.com/cis/2017/07/sap-%D0%BF%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%B8%D0%BB-%D0%BD%D0%BE%D0%B2%D1%8B%D0%B5-iot-%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81%D1%8B-%D0%BD%D0%B0-%D0%BA%D0%BE%D0%BD%D1%84%D0%B5%D1%80%D0%B5/>.
5. Облачная платформа SAP — Введение [Электрон. ресурс]. – Режим доступа: <https://coderlessons.com/tutorials/sap/izuchite-sap-cloud-platform/oblachnaia-platforma-sap-kratkoe-rukovodstvo>.
6. SAP Cloud Platform - облачная CRM-система [Электрон. ресурс]. – Режим доступа: <https://atlantconsult.ru/technology/sap-cloud-platform/>.
7. SAP Cloud Platform. Что такое облачная платформа SAP? [Электрон. ресурс]. – Режим доступа: [https://dataseed.ru/produkty/sap-cloud-platform-chto-takoe-oblachnaya-platforma-sap/#SAP\\_Cloud\\_Platform\\_SDK\\_iOS](https://dataseed.ru/produkty/sap-cloud-platform-chto-takoe-oblachnaya-platforma-sap/#SAP_Cloud_Platform_SDK_iOS).
8. Запуск приложений с блокчейн на SAP Cloud Platform [Электрон. ресурс]. – Режим доступа: <https://habr.com/ru/company/sap/blog/347556/>.
9. DevOps: Breaking the Development-Operations barrier [Electronic resource]. – Access mode: <https://www.atlassian.com/ru/devops/what-is-devops>.

10. DevOps [Электрон. ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/DevOps>.

11. CALMS (ex-CAMS) — основа DevOps [Электрон. ресурс]. – Режим доступа: <http://agilemindset.ru/calms/>.

12. Гачко Д. Что такое методология DevOps и кому она нужна [Электрон. ресурс] / Д. Гачко. – Режим доступа: <https://vc.ru/hr/50165-что-такое-metodologiya-devops-i-komu-ona-nuzhna>.

13. Почему я не люблю DevOps (и современное ПО) [Электрон. ресурс]. – Режим доступа: <https://habr.com/ru/post/349796/>.

14. Вичугова, А. DevOps [Электрон. ресурс] / А. Вичугова. – Режим доступа:

[https://www.bigdataschool.ru/wiki/devops#:~:text=DevOps%20\(Development%20Operation\)%20E2%80%93%20D1%8D%D1%82%D0%BE,%D1%81%D0%BF%D0%B5%D1%86%D0%B8%D0%B0%D0%BB%D0%B8%D1%81%D1%82%D0%BE%D0%B2%20D1%81%20D0%BF%D0%BE%D0%BC%D0%BED1%89%D1%8C%D1%8E%20D0%B8%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%B2%20D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8](https://www.bigdataschool.ru/wiki/devops#:~:text=DevOps%20(Development%20Operation)%20E2%80%93%20D1%8D%D1%82%D0%BE,%D1%81%D0%BF%D0%B5%D1%86%D0%B8%D0%B0%D0%BB%D0%B8%D1%81%D1%82%D0%BE%D0%B2%20D1%81%20D0%BF%D0%BE%D0%BC%D0%BED1%89%D1%8C%D1%8E%20D0%B8%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%B2%20D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8).

15. SAP HANA ENTERPRISE Cloud <https://roi4cio.com/catalog/product/sap-hana-enterprise-cloud>.

16. SAP HCP: HANA Cloud Platform <http://www.itctg.ru/solutions/sap-hana-cloud>.

17. Табаков, В. HANA Enterprise Cloud - бизнес-приложения SAP на HANA в облаке [Электрон. ресурс] / В. Табаков. – Режим доступа: <https://sapland.ru/kb/blogs/hana-enterprise-cloud-biznes-prilozheniya-sap-na-hana-v-oblake.html>.

18. SAP S/4 HANA - новое поколение комплекса корпоративных приложений SAP Business Suite [Электрон. ресурс]. – Режим доступа: <http://www.itctg.ru/solutions/s4-hana-sap-business-suite>.

19. Залюбовський Я.І. Хмарні рішення SAP для реалізації технологій Інтернету речей / Я.І. Залюбовський, А.В. Пархоменко, О.А.Поздняков // Тиждень науки: Щорічна НПК викладачів, науковців, молодих учених, аспірантів та студентів, Запоріжжя, 15-19 квітня 2019 року: тези доповідей. - Запоріжжя: ЗНТУ, 2019. - С.79.
20. Как перестать бояться и начать разрабатывать на SAP Cloud Platform [Электрон. ресурс]. – Режим доступа: <https://habr.com/ru/article/323088/>.
21. АВАР/4 [Электрон. ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/АВАР/4>.
22. Кречмер, Р. Разработка приложений SAP R/3 на языке АВАР/4 / Р.Кречмер, В. Вейс. – М.: Лори, 2014. - 484 с.
23. Введение в АВАР-инструментальные средства. Руководство для участников курса. - SAP AG, 2006. - 421 с.
24. Разработка мультитенантных приложений на SAP Cloud Platform в среде Нео, часть 1 [Электрон. ресурс]. – Режим доступа: <https://habr.com/ru/company/sap/blog/415351/>.
25. Как с помощью SAP Cloud Platform создать расширение для цифрового ядра [Электрон. ресурс]. – Режим доступа: <https://habr.com/ru/company/sap/blog/502572/>.
26. Руководство по разработке облачных приложений с помощью SAP Cloud Platform и Cloud Foundry [Электрон. ресурс]. – Режим доступа: <https://habr.com/ru/company/sap/blog/350690/>.
27. Остапенко, В.В. Методичні вказівки по економічному обґрунтуванню дипломних проектів для студентів спеціальностей 8.05010301 «Програмне забезпечення автоматизованих систем» та 8.05010201 «Інформаційні технології проектування» для усіх форм навчання / В.В. Остапенко, С.М Сердюк. – Запоріжжя: ЗНТУ, 2015.– 22 с.
28. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПіН

3.3.2.007-98. – [Чинний від 1998-12-10]. – К.: МОЗ України, 1998. – 20 с. (Державні санітарні правила та норми).

29. Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги: ДСТУ 8604:2015. - [Чинний від 2017-07-01]. – К.: Держстандарт України, 2017. – 20 с. (Національний стандарт України).

30. Правила улаштування електроустановок : ПУЕ-2017. – На заміну ПУЕ-86. - [Чинний від 2017-08-21]. – К. : Міненерговугілля України, 2017. – 617 с.

31. Система стандартів безпеки праці. Електробезпека. Загальні вимоги та номенклатура видів захисту: ДСТУ 7237:2011. - [Чинний від 2011-08-01]. – Київ: Держспоживстандарт України, 2011. – 9 с. (Національний стандарт України).

32. Правила будови електроустановок. Електрообладнання спеціальних установок: НПАОП 40.1-1.32-01. - [Чинний від 2001-06-21]. - Харків: Індустрія, 2010. – 118 с. (Державний нормативний акт про охорону праці).

33. Захист проти ураження електричним струмом. Загальні аспекти щодо установок та обладнання: ДСТУ 61140:2015. - [Чинний від 2001-06-21]. - Київ: Держспоживстандарт України, 2015. – 35 с. (Національний стандарт України)

34. Основи професійної безпеки та здоров'я людини: підручник / В. В. Березуцький [та ін.] ; під ред. проф. В. В. Березуцького. – Харків : НТУ ХПІ, 2018. – 553 с.

35. Природне та штучне освітлення: ДБН В.2.5-28:2018. - [Чинні від 2019-03-01]. - К: Мінрегіон України, 2018. – 133 с. (Державні будівельні норми України).

36. Санітарні норми мікроклімату виробничих приміщень: ДСН 3.3.6.042-99. - [Чинні від 1999-12-01]. – К.: МОЗ, 2000. – 10 с. (Державні санітарні норми).

37. Жидецький, В. Ц. Основи охорони праці: підручник / В. Ц. Жидецький. – 5-те вид., доп. – К.: Знання, 2014. – 373 с.
38. Пожежна безпека об'єктів будівництва. Загальні вимоги: ДБН В.1.1-7:2016. - [Чинні від 2017-02-01]. – Київ: Міністерство регіонального розвитку, будівництва та житлово-комунального господарства України, 2017. – 39 с. (Державні будівельні норми).
39. Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою : ДСТУ Б В.1.1-36:2016. – На заміну НАПБ Б.03.002-2007. - [Чинний від 2017-01-01]. – К. : Мінрегіонбуд України, 2016. – 66 с. (Національний стандарт України)
40. Стеблюк М. І. Цивільна оборона та цивільний захист: навч. посіб. для вузів / М. І. Стеблюк. – К. : Знання, 2013. – 487 с.

**ДОДАТОК А**  
**Текст програми**

## A1. ProgramCode\_GroupPersonalOrder

```

*&-----
_*
*& Report Z_BPO03_GRP_PRINT_PROG
*&-----*
*&
*&-----*
REPORT z_bpo03_grp_print_prog.

INCLUDE z_bpo03_grp_prprg_top.
INCLUDE z_bpocomm_prprg_selscr ##INCL_OK.
INCLUDE z_bpo03_grp_prprg_events.
INCLUDE z_bpocomm_prprg_forms ##INCL_OK.
INCLUDE z_bpo03_grp_prprg_forms.

*&-----
_*
*& Include          Z_BPO03_GRP_PRPRG_TOP
*&-----*
INCLUDE z_bpocomm_prprg_top ##INCL_OK.

INFOTYPES: 0007, 0008, 9005.

TYPES: BEGIN OF ts_pers_data,
        A04          TYPE string,
        A05          TYPE string,
        A07          TYPE string,
        A08          TYPE string,
        A09          TYPE string,
        A10          TYPE string,
        A11          TYPE string,
        A12          TYPE string,
        A13          TYPE string,
        A14          TYPE string,
        A15          TYPE string,
        A16          TYPE string,
        A17          TYPE string,
        A18          TYPE string,
        A19          TYPE string,
        A20          TYPE string,
*          stabs_objid TYPE p1000-objid,

```

```

*          stabs_stext  TYPE p1000-stext,
*          fname_n     TYPE string,
          END OF ts_pers_data.

```

```

TYPES: tt_pers_data TYPE TABLE OF ts_pers_data WITH EMPTY KEY.

```

```

*TYPES: BEGIN OF ts_sign_data,
*          A28 TYPE string,
*          END OF ts_sign_data.

```

```

*TYPES: tt_sign_data TYPE TABLE OF ts_sign_data WITH EMPTY KEY.

```

```

*

```

```

*TYPES: BEGIN OF ts_stabx_data,
*          A27          TYPE string,
*          TAB_SIGN     TYPE tt_sign_data,
*          END OF ts_stabx_data.

```

```

*TYPES: tt_stabx_data TYPE TABLE OF ts_stabx_data WITH EMPTY KEY.

```

```

TYPES: BEGIN OF ts_templ_data,

```

```

          LOGO          TYPE xstring,
          SH01          TYPE string,
          SH02          TYPE string,
          SH03          TYPE string,
          SH04          TYPE string,
          SH05          TYPE string,
          A01           TYPE string,
          A03           TYPE string,
          CHPER_6_PLANSLTEXTD TYPE string,
          CHPER_6_CHNAMD  TYPE string,
          CHPER_1_PLANSLTEXTN TYPE string,
          CHPER_1_SIGNAMN TYPE string,
*          A26           TYPE string,
          TAB_PERS      TYPE tt_pers_data,
*          TAB_STABX    TYPE tt_stabx_data,
          EXEC_NACHNN   TYPE string,
          EXEC_WORKINTPHONE TYPE string,
          stabs_str     TYPE string,
          tab_stabs_reader TYPE zcl_bpocomm_static=>tt_stabs_reader,
          END OF ts_templ_data.

```

```

DATA: gs_templ_data TYPE ts_templ_data ##NEEDED.

```

```

DATA gv_is_first TYPE abap_bool.

```

```

*&-----

```

```

-*

```

```
*& Include Z_BPOCOMM_PRPRG_SELSCR
```

```
*&-----*
```

```
SELECTION-SCREEN BEGIN OF BLOCK b11.
```

```
PARAMETERS:
```

```
p_lang TYPE sy-langu DEFAULT sy-langu,
```

```
ftype TYPE t7rur0-fotype,
```

```
form TYPE t7rur0-namef,
```

```
massn TYPE massn,
```

```
massg TYPE massg,
```

```
ordnu TYPE p0298-ordnu,
```

```
orddt TYPE p0298-orddt,
```

```
maspr TYPE char1,
```

```
armod TYPE syarmod,
```

```
arver TYPE p33_arver,
```

```
pv_guid TYPE guid_32.
```

```
SELECTION-SCREEN END OF BLOCK b11.
```

```
*&-----*
```

```
_*
```

```
*& Include Z_BPO03_GRP_PRPRG_EVENTS
```

```
*&-----*
```

```
INITIALIZATION.
```

```
START-OF-SELECTION.
```

```
PERFORM reinit_sel_data_interval.
```

```
PERFORM init_predef_fields.
```

```
PERFORM start_of_selection.
```

```
GET peras.
```

```
PERFORM actualize_p0298.
```

```
PERFORM add_data_for_predef_fields.
```

```
PERFORM get_peras.
```

```
END-OF-SELECTION.
```

```
PERFORM fill_predef_fields CHANGING gs_tmpl_data.
```

```
PERFORM call_form.
```

```

*&-----
-*
*& Include          Z_BPOCOMM_PRPRG_FORMS
*&-----*

*&-----*
*& Form REINIT_SEL_PERAMS
*&-----*

FORM reinit_sel_data_interval.
  IF pnpnpernr[] IS INITIAL.
    RETURN.
  ENDIF.

  TRY.
    DATA(lt_p0298) = zcl_bhrcomm_static=>get_single_pernr_it0298_tab(
      EXPORTING
        iv_pernr = pnpnpernr[ 1 ]-low
        iv_begda = pnpbegda
        iv_endda = pnpendda ).
    CATCH zcx_bhrcomm. " Виключні подій пакета Z_BHR_COMMON
      RETURN.
    ENDTRY.

  READ TABLE lt_p0298 INTO DATA(ls_p0298)
    WITH KEY massn = massn massg = massg
           ordnu = ordnu orddt = orddt.

  IF sy-subrc <> 0.
    RETURN.
  ENDIF.

  IF ls_p0298-grpor = abap_false.
    RETURN.
  ENDIF.

  DATA lt_pnpnpernr LIKE pnpnpernr[].
  CALL FUNCTION 'HR_RU_PERSONS_FROM_ORDER'
    EXPORTING
      p_p0298 = ls_p0298
    IMPORTING
      begda   = pn-begda
      endda   = pn-endda
    TABLES
      pnr     = lt_pnpnpernr.

```

ENDFORM.

```
*&-----*
*&      Form  ACTUALIZE_P0298
*&-----*
```

FORM actualize\_p0298 .

DATA (lt\_p0298) = p0298[].

DELETE p0298 WHERE massn <> massn

OR massg <> massg

OR ordnu <> ordnu

OR orddt <> orddt.

IF p0298[] IS INITIAL.

p0298[] = lt\_p0298.

DELETE p0298 WHERE massn <> massn

OR ordnu <> ordnu

OR orddt <> orddt.

ENDIF.

rp\_provide\_from\_last p0298 space pn-begda pn-endda.

ENDFORM.

```
*&-----*
*&      Form  INIT_PREDEF_FIELDS
*&-----*
```

FORM init\_predef\_fields.

CLEAR go\_predef\_fields.

go\_predef\_fields = NEW #( ).

go\_predef\_fields->set\_pref\_pernr( iv\_pref\_pernr = pnppernr-low ).

ENDFORM.

```
*&-----*
*&      Form  add_data_for_predef_fields
*&-----*
```

FORM add\_data\_for\_predef\_fields .

IF go\_predef\_fields IS NOT BOUND. RETURN. ENDIF.

rp\_provide\_from\_last p0001 space p0298-begda p0298-begda.

rp\_provide\_from\_last p0002 space p0298-begda p0298-begda.

go\_predef\_fields->add\_pernr\_data(

is\_pernr\_data = VALUE #( pernr = p0298-

```
pernr p0001 = p0001 p0002 = p0002 p0298 = p0298 ) ).
```

```
ENDFORM.
```

```
*&-----*
```

```
*&      Form  FILL_PREDEF_FIELDS
```

```
*&-----*
```

```
FORM fill_predef_fields
```

```
  CHANGING cs_templ_data TYPE data.
```

```
  IF go_predef_fields IS NOT BOUND. RETURN. ENDIF.
```

```
  go_predef_fields-
```

```
>fill_predef_fields( CHANGING cs_templ_data = cs_templ_data ).
```

```
ENDFORM.
```

```
*&-----*
```

```
*&      Form  INIT_FORM
```

```
*&-----*
```

```
*      text
```

```
*-----*
```

```
FORM init_form
```

```
  CHANGING co_view TYPE REF TO zif_bpocomm_view.
```

```
  TRY.
```

```
    CLEAR co_view.
```

```
    co_view = zcl_bpocomm_static=>get_view_instance( iv_view_class_name = C
ONV #( form ) ).
```

```
    "PERFORM set_view_parameters IN PROGRAM uv_set_view_param_prog USING lo
_view IF FOUND.
```

```
    "lo_view->show( id_templ_data = us_templ_data ).
```

```
  CATCH cx_static_check INTO DATA(lo_error).
```

```
    WHILE lo_error->previous IS BOUND.
```

```
      lo_error = CAST #( lo_error->previous ).
```

```
    ENDWHILE.
```

```
    MESSAGE lo_error->get_text( ) TYPE 'I' DISPLAY LIKE 'E'.
```

```
  ENDTRY.
```

```
ENDFORM.
```

```
*&-----*
```

```
*&      Form  SHOW_FORM
```

```
*&-----*
```

```
*      text
```

```
*-----*
```

```

FORM show_form
  USING co_view TYPE REF TO zif_bpocomm_view
        us_tmpl_data TYPE data.
  IF co_view IS NOT BOUND.
    RETURN.
  ENDIF.

  TRY.
    co_view->show( id_tmpl_data = us_tmpl_data ).
  CATCH cx_static_check INTO DATA(lo_error).
    WHILE lo_error->previous IS BOUND.
      lo_error = CAST #( lo_error->previous ).
    ENDWHILE.
    MESSAGE lo_error->get_text( ) TYPE 'I' DISPLAY LIKE 'E'.
  ENDTRY.
ENDFORM.

*&-----
--*
*&  Include          Z_BPO03_GRP_PRPRG_FORMS
*&-----*
*&-----*
*&      Form  START_OF_SELECTION
*&-----*
FORM start_of_selection .
  gv_is_first = abap_true.
  CLEAR gs_tmpl_data.
ENDFORM.

*&-----*
*&      Form  GET_PERAS
*&-----*
FORM get_peras.
  IF gv_is_first = abap_true.
    PERFORM get_from_p0298.

    gv_is_first = abap_false.
  ENDIF.
  APPEND INITIAL LINE TO gs_tmpl_data-tab_pers ASSIGNING FIELD-
  SYMBOL(<ls_pers_data>).
  PERFORM get_from_p0001g CHANGING <ls_pers_data>.
  PERFORM get_from_p0002g CHANGING <ls_pers_data>.
  PERFORM get_from_p0007g CHANGING <ls_pers_data>.
  PERFORM get_from_p0008g CHANGING <ls_pers_data>.

```

```

PERFORM get_from_p0298g CHANGING <ls_pers_data>.

ENDFORM.

*&-----*
*&      Form  CALL_FORM
*&-----*

FORM call_form.
  DATA lo_view TYPE REF TO zif_bpocomm_view.
  PERFORM init_form CHANGING lo_view.
  PERFORM set_form_param USING lo_view.
  PERFORM show_form USING lo_view gs_tmpl_data.
ENDFORM.

*&-----*
*&      Form  SET_VIEW_PARAMETERS
*&-----*

FORM set_form_param
USING uo_view TYPE REF TO zif_bpocomm_view.
  uo_view->set_parameter(
    EXPORTING iv_param_nam = 'FILE_NAME'
    iv_param_val = |{ p0298-
orddt DATE = ENVIRONMENT }_{ zcl_bhrcomm_static=>get_it0002_sname( p0002 ) }_|
    { p0298-ordnu }| ).
ENDFORM.

*&-----*
*&      Form  GET_FROM_P0298
*&-----*

FORM get_from_p0298.
  rp_provide_from_last p0001 space p0298-begda p0298-endda.
  IF p0298-massn = 39.
    gs_tmpl_data-a01 = TEXT-011.
    gs_tmpl_data-a03 = TEXT-031.
  ENDIF.
ENDFORM.

*&-----*
*&      Form  GET_FROM_P0001g
*&-----*

FORM get_from_p0001g CHANGING cs_pers_data TYPE ts_pers_data.
  rp_provide_from_last p0001 space p0298-begda p0298-endda.
  rp_provide_from_last p0002 space p0298-begda p0298-endda.
  rp_provide_from_last p9005 space p0298-begda p0298-endda.

*begin -----GET A05-----
  TRY.

```

```

DATA(ls_p0001) = zcl_bhrcomm_static=>get_single_pernr_it0001(
    iv_pernr = p0298-pernr
    iv_ondat = CONV #( p0298-begda - 1 ) ).
CATCH zcx_bhrcomm.
ENDTRY.
TRY.
    cs_pers_data-a05 = zcl_bhrcomm_static=>get_single_plans_ltext(
    iv_plans = ls_p0001-plans
    iv_ondat = CONV #( p0298-begda - 1 )
    iv_subty = '9002'
    iv_first_symb_case = zcl_bhrcomm_static=>mc_case_lower ).
CATCH zcx_bhrcomm.
ENDTRY.
TRY.
DATA(ls_p9005) = zcl_bhrcomm_static=>get_single_pernr_it9005(
    iv_pernr = p0298-pernr
    iv_ondat = CONV #( p0298-begda - 1 ) ).
    cs_pers_data-a05 = ||{ cs_pers_data-
a05 } { zcl_bhrcomm_static=>get_single_it9005_qual_text( ls_p9005-
qualification ) }|.
CATCH zcx_bhrcomm.
ENDTRY.
*end -----GET A05-----
*begin -----GET A07-----
TRY.
DATA(lt_rh_struct) = zcl_bhrcomm_static=>get_rh_up_o_struct(
EXPORTING iv_orgeh = ls_p0001-orgeh iv_ondat = ls_p0001-begda ).

DATA(lt_rh_object_ltext) = zcl_bhrcomm_static=>get_rh_struct_ltext(
EXPORTING
    it_struct = lt_rh_struct
    iv_subty = '9002'
    iv_first_symb_case = zcl_bhrcomm_static=>mc_case_lower ).
CATCH zcx_bhrcomm.
ENDTRY.
SORT lt_rh_object_ltext BY level ASCENDING.

TRY.
DATA(ls_p1000) = zcl_bhrcomm_static=>get_rh_stabs(
EXPORTING
    iv_otype = zcl_bhrcomm_static=>mc_otype_o
    iv_objid = ls_p0001-orgeh " ID объекта
    iv_ondat = CONV #( p0298-begda - 1 ) ).

```

```

    CATCH zcx_bhrcomm.
ENDTRY.
LOOP AT lt_rh_object_ltext ASSIGNING FIELD-SYMBOL(<ls_rh_object_ltext>).
    cs_pers_data-a07 = |{ cs_pers_data-a07 } { <ls_rh_object_ltext>-ltext }|.
    IF <ls_rh_object_ltext>-objid = ls_p1000-objid .
        EXIT.
    ENDIF.
ENDLOOP.
UNASSIGN <ls_rh_object_ltext>.
CLEAR lt_rh_struct.
*end -----GET A07-----
*begin -----GET A08-----
    ls_p0001 = zcl_bhrcomm_static=>get_single_pernr_it0001(
EXPORTING iv_pernr = p0298-pernr iv_ondat = CONV #( p0298-begda - 1 ) ).
    IF ls_p0001-werks+2(2) = '01'.
        cs_pers_data-a08 = TEXT-081.
    ELSEIF ls_p0001-werks+2(2) = '02'.
        cs_pers_data-a08 = TEXT-082.
    ENDIF.
*end -----GET A08-----
*begin -----GET A10-----
    TRY.
        cs_pers_data-a10 = zcl_bhrcomm_static=>get_single_plans_ltext(
            iv_plans = p0001-plans
            iv_ondat = p0298-begda
            iv_subty = '9002' ).
        CATCH zcx_bhrcomm.
    ENDTRY.
    TRY.
        cs_pers_data-a10 = |{ cs_pers_data-
a10 } { zcl_bhrcomm_static=>get_single_it9005_qual_text( p9005-
qualification ) }|.
        CATCH zcx_bhrcomm.
    ENDTRY.
*end -----GET A10-----
*begin -----GET A11-----
    TRY.
        lt_rh_struct = zcl_bhrcomm_static=>get_rh_up_o_struct(
            EXPORTING iv_orgeh = p0001-orgeh iv_ondat = p0001-begda ).
        CATCH zcx_bhrcomm.
    ENDTRY.
    SORT lt_rh_object_ltext BY level ASCENDING.

```

```

LOOP AT lt_rh_object_ltext ASSIGNING <ls_rh_object_ltext>.
  cs_pers_data-a11 = |{ cs_pers_data-a11 } { <ls_rh_object_ltext>-ltext }|.
  IF <ls_rh_object_ltext>-objid = ls_p1000-objid .
    EXIT.
  ENDIF.
ENDLOOP.
UNASSIGN <ls_rh_object_ltext>.
CLEAR lt_rh_struct.
*end -----GET A11-----
*begin -----GET A12-----
  IF p0001-werks+2(2) = '01'.
    cs_pers_data-a12 = TEXT-121.
  ELSEIF p0001-werks+2(2) = '02'.
    cs_pers_data-a12 = TEXT-122.
  ENDIF.
*end -----GET A12-----
*begin -----GET A13-----
  IF p0001-endda <> '99991231'.
    cs_pers_data-a13 = |{ TEXT-131 } { p0001-
begda DATE = ENVIRONMENT } { TEXT-132 } { p0001-endda DATE = ENVIRONMENT }|.
  ELSE.
    cs_pers_data-a13 = |{ TEXT-131 } { p0001-begda DATE = ENVIRONMENT }|.
  ENDIF.
*end -----GET A13-----
*begin -----GET A15 A16-----
  IF p0298-massn = '02' AND ( p0298-masss = '01' OR p0298-masss = '02').
    DATA lt_objecttab TYPE hrobject_t.
    DATA lt_ex_1001_tab TYPE p1001tab.
    APPEND INITIAL LINE TO lt_objecttab ASSIGNING FIELD-
SYMBOL(<ls_objecttab>).
    <ls_objecttab>-plvar = zcl_bhrcomm_static=>get_active_plvar( ).
    <ls_objecttab>-otype = zcl_bhrcomm_static=>mc_otype_s.
    <ls_objecttab>-objid = p0001-plans.
    UNASSIGN <ls_objecttab>.
    CALL FUNCTION 'HR_HCP_GET_EMPL_FOR_POSITION'
      EXPORTING
        im_objecttab      = lt_objecttab
        im_begda          = p0298-begda
        im_endda          = p0298-begda
      IMPORTING
        ex_1001_tab      = lt_ex_1001_tab
    EXCEPTIONS
      read_1001_error = 1

```

```

        OTHERS          = 2.
IF sy-subrc = 0.
    LOOP AT lt_ex_1001_tab ASSIGNING FIELD-SYMBOL(<ls_ex_1001_tab>).
        IF <ls_ex_1001_tab>-sobid <> p0298-pernr.
            TRY.
                DATA(ls_p0002) = zcl_bhrcomm_static=>get_single_pernr_it0002(
                    iv_pernr = CONV #( <ls_ex_1001_tab>-sobid )
                    iv_ondat = p0298-begda ).
                cs_pers_data-a15 = zcl_bhrcomm_static=>get_it0002_chnam(
                    iv_noun   = zcl_bhrcomm_static=>mc_noun_r
                    is_p0002 = ls_p0002 ).
                CATCH zcx_bhrcomm.
            ENDTRY.
            cs_pers_data-a16 = <ls_ex_1001_tab>-sobid.
        ENDIF.
    ENDLOOP.
ENDIF.
ENDIF.
ENDFORM.

*&-----*
*&      Form  GET_FROM_P0002g
*&-----*
FORM get_from_p0002g CHANGING cs_pers_data TYPE ts_pers_data.
    rp_provide_from_last p0002 space p0298-begda p0298-endda.

    cs_pers_data-a04 = zcl_bhrcomm_static=>get_it0002_fname(
        iv_noun   = zcl_bhrcomm_static=>mc_noun_z
        is_p0002 = p0002 ).
ENDFORM.

*&-----*
*&      Form  GET_FROM_P0298G
*&-----*
FORM get_from_p0298g CHANGING cs_pers_data TYPE ts_pers_data.
    cs_pers_data-a09 = p0298-pernr.
*begin -----GET A14-----
    IF p0298-massg = '01'.
        cs_pers_data-a14 = TEXT-141.
    ELSE.
        TRY.
            zcl_bhrcomm_static=>get_single_masss_texts(
                EXPORTING
                    iv_massn      = p0298-massn          " Тип операції
                    iv_massg      = p0298-massg          " Причина операції

```

```

        iv_masss      = p0298-
masss              " Подробиці причин для кадрових заходів
        iv_ondat     = p0298-begda              " Системна дата
        IMPORTING
        ev_mltxt_long = DATA(lv_mltxt_long)    " Подробиці про причини кадрових заходів (довгий_текст)
    ).
        cs_pers_data-a14 = lv_mltxt_long.
        CATCH zcx_bhrcomm.
    ENDTRY.
ENDIF.
*end -----GET A14-----
*begin -----GET A19-----
    TRY.
        DATA(lt_p1001) = zcl_bhrcomm_static=>get_single_objec_it1001(
        EXPORTING iv_otype = zcl_bhrcomm_static=>mc_otype_s
            iv_objid = CONV #( p0001-plans )
            iv_subty = 'B003'
            iv_ondat = p0298-begda ).
        DATA(ls_p1001) = VALUE #( lt_p1001[ sclas = zcl_bhrcomm_static=>mc_otype_s ]
        e_a ] DEFAULT VALUE p1001( ) ).
        DATA(ls_p1640) = zcl_bhrcomm_static=>get_single_objec_it1640(
        EXPORTING iv_objid = CONV #( ls_p1001-sobid )
            iv_ondat = p0298-begda ).
        CATCH zcx_bhrcomm.
    ENDTRY.
    IF ls_p1640 IS NOT INITIAL.
        IF ls_p1640-lbcond = 0.
            cs_pers_data-a19 = TEXT-195.
        ELSE.
            cs_pers_data-a19 = TEXT-190.
            cs_pers_data-a19 = replace( val = cs_pers_data-a19 sub = '&1' with = |{ conv i( ls_p1640-addpay ) }| ).
            cs_pers_data-a19 = SWITCH #( ls_p1640-lbcond
                WHEN '1' THEN replace( val = cs_pers_data-a19 sub = '&2' with = |{ TEXT-191 }| )
                WHEN '2' THEN replace( val = cs_pers_data-a19 sub = '&2' with = |{ TEXT-192 }| )
                WHEN '3' THEN replace( val = cs_pers_data-a19 sub = '&2' with = |{ TEXT-193 }| )
                WHEN '4' THEN replace( val = cs_pers_data-a19 sub = '&2' with = |{ TEXT-194 }| )
            ELSE replace( val = cs_pers_data-a19 sub = '&2' with = | | ) ).

```

```

        cs_pers_data-a19 = replace( val = cs_pers_data-
a19 sub = '&3' with = |{ ls_p1640-lstnum }| ).
        cs_pers_data-a19 = replace( val = cs_pers_data-
a19 sub = '&4' with = |{ ls_p1640-zz_lstnum }| ).
        ENDIF.
    ENDIF.
*end -----GET A19-----
ENDFORM.
*&-----*
*&      Form  GET_FROM_P0008G
*&-----*
FORM get_from_p0008g CHANGING cs_pers_data TYPE ts_pers_data.
    rp_provide_from_last p0008 space p0298-begda p0298-endda.

    TRY.
        cs_pers_data-
a17 = |{ conv i( zcl_bhrcomm_static=>get_it0008_lgart_pbwla(
            EXPORTING is_p0008 = p0008 iv_lgart = '1000' )-betrq )|}.
        CATCH zcx_bhrcomm.
    ENDTRY.
*begin -----GET A18 -----
    TRY.
        DATA(lt_lgart_data) = zcl_bhrcomm_static=>get_it0008_lgarts_pbwla( is_p
0008 = p0008 ).
        IF lt_lgart_data IS NOT INITIAL.
            DATA: lt_lgart_types TYPE RANGE OF lgart.
            lt_lgart_types = VALUE #( sign = 'I' option = 'EQ' ( low = '2030' ) (
low = '2060' )
                ( low = '2070' ) ( low = '2080' ) ( low = '2090' ) ( low = '2110' ) (
low = '2150' )
                ( low = '2160' ) ( low = '2170' ) ( low = '2290' ) ( low = '2500' ) (
low = '2510' )
                ( low = '2520' ) ( low = '2530' ) ( low = '2540' ) ( low = '2550' ) (
low = '2580' )
                ( low = '2590' ) ( low = '2600' ) ( low = '2610' ) ).
        TRY.
            DATA(lv_molga) = zcl_bhrcomm_static=>get_country_grouping( iv_per
sa = p0001-werks ).
            CATCH zcx_bhrcomm.
        ENDTRY.
        LOOP AT lt_lgart_data ASSIGNING FIELD-SYMBOL(<ls_lgart_data>).
            DATA(lv_num_part) = VALUE string( ).

```

```

IF <ls_lgart_data>-lgart IN lt_lgart_types.
  lv_num_part = |{ conv i( <ls_lgart_data>-anzhl ) }|{ TEXT-181 }|.
ELSEIF <ls_lgart_data>-lgart = '2570'.
  lv_num_part = |{ conv i( <ls_lgart_data>-betrg ) }|{ TEXT-182 }|.
ENDIF.

IF lv_num_part IS NOT INITIAL
OR <ls_lgart_data>-lgart = '1400'.
  TRY.
    DATA(lv_lgart_text) = zcl_bhrcomm_static=>get_single_lgart_lt
ext(
      iv_lgart = <ls_lgart_data>-lgart iv_molga = lv_molga ).
    CATCH zcx_bhrcomm.
  ENDTRY.
  DATA(lv_row_text) = |{ lv_num_part } { lv_lgart_text }|.
  CONDENSE lv_row_text.
ENDIF.
IF lv_row_text IS NOT INITIAL.
  IF cs_pers_data-a18 IS INITIAL.
    cs_pers_data-a18 = lv_row_text.
  ELSE.
    cs_pers_data-a18 = |{ cs_pers_data-a18 }, { lv_row_text }|.
  ENDIF.
ENDIF.
ENDLOOP.
UNASSIGN <ls_lgart_data>.
ENDIF.
CATCH zcx_bhrcomm.
ENDTRY.
*end -----GET A18 -----
ENDFORM.
*&-----*
*&      Form  GET_FROM_P0007G
*&-----*
FORM get_from_p0007g CHANGING cs_pers_data TYPE ts_pers_data.
  rp_provide_from_last p0007 space p0298-begda p0298-endda.

  TRY.
    cs_pers_data-a20 = zcl_bhrcomm_static=>get_schkz_text(
      EXPORTING is_p0001 = p0001 iv_schkz = p0007-schkz ).
    CATCH zcx_bhrcomm.
  ENDTRY.
ENDFORM.

```

## A2. ProgramCode\_IndividualPersonalOrder

```

*&-----
-*
*& Report Z_BPO03_PRINT_PROG
*&-----*
*&
*&-----*
REPORT z_bpo03_print_prog.

INCLUDE z_bpo03_prprg_top.
INCLUDE z_bpocomm_prprg_selscr ##INCL_OK.
INCLUDE z_bpo03_prprg_events.
INCLUDE z_bpocomm_prprg_forms ##INCL_OK.
INCLUDE z_bpo03_prprg_forms.

*&-----
-*
*& Include          Z_BPO03_PRPRG_TOP
*&-----*
INCLUDE z_bpocomm_prprg_top ##INCL_OK.

INFOTYPES: 0007, 0008, 0016, 9005.

TYPES: BEGIN OF ts_a13,
        a13 TYPE string,
      END OF ts_a13,
      tt_a13 TYPE TABLE OF ts_a13 WITH EMPTY KEY.
TYPES: BEGIN OF ts_a14_1,
        a14_1 TYPE string,
      END OF ts_a14_1,
      tt_a14_1 TYPE TABLE OF ts_a14_1 WITH EMPTY KEY.
TYPES: BEGIN OF ts_a16,
        a16 TYPE string,
      END OF ts_a16,
      tt_a16 TYPE TABLE OF ts_a16 WITH EMPTY KEY.
TYPES: BEGIN OF ts_a17,
        a17 TYPE string,
      END OF ts_a17,
      tt_a17 TYPE TABLE OF ts_a17 WITH EMPTY KEY.
TYPES: BEGIN OF ts_a19,
        a19 TYPE string,
      END OF ts_a19,
      tt_a19 TYPE TABLE OF ts_a19 WITH EMPTY KEY.

```

```

TYPES: BEGIN OF ts_a20,
        a20 TYPE string,
END OF ts_a20,
tt_a20 TYPE TABLE OF ts_a20 WITH EMPTY KEY.

```

```

TYPES: BEGIN OF ts_templ_data,
        sh01          TYPE string,
        a62           TYPE string,
        a06_1         TYPE string,
        a06_2         TYPE string,
        sh04          TYPE string,
        sh02          TYPE string,
        a4            TYPE string,
        a5            TYPE string,
        a6            TYPE string,
        a7            TYPE string,
        a8            TYPE string,
        a9            TYPE string,
        a10           TYPE string,
        a11           TYPE string,
        a12           TYPE string,
        tab_a13        TYPE tt_a13,
        tab_a14_1      TYPE tt_a14_1,
        a61           TYPE string,
        a14           TYPE string,
        a15           TYPE string,
        tab_a16        TYPE tt_a16,
        tab_a17        TYPE tt_a17,
        tab_a19        TYPE tt_a19,
        tab_a20        TYPE tt_a20,
        a17           TYPE string,
        a18           TYPE string,
        a19           TYPE string,
        a20           TYPE string,
        a21           TYPE string,
        a22           TYPE string,
        a23           TYPE string,
        a24           TYPE string,
        a25           TYPE string,
        a26           TYPE string,
        a27           TYPE string,
        a28           TYPE string,
        a29           TYPE string,

```

```

a31          TYPE string,
a32          TYPE string,
a33          TYPE string,
a34          TYPE string,
a35          TYPE string,
a36          TYPE string,
a37          TYPE string,
a38          TYPE string,
a39          TYPE string,
a40          TYPE string,
a41          TYPE string,
a42          TYPE string,
a43          TYPE string,
a44          TYPE string,
a45          TYPE string,
a46_1        TYPE string,
a46_2        TYPE string,
a46_3        TYPE string,
a46_4        TYPE string,
a47_1        TYPE string,
a47_2        TYPE string,
a47_3        TYPE string,
a47_4        TYPE string,
a48          TYPE string,
a49          TYPE string,
stabs_str    TYPE string,
exec_nachnn  TYPE string,
exec_workintphone TYPE string,
END OF ts_templ_data.

```

```
DATA: gs_templ_data TYPE ts_templ_data ##NEEDED.
```

```

*&-----
-*
*& Include          Z_BPOCOMM_PRPRG_SELSCR
*&-----*

```

```
SELECTION-SCREEN BEGIN OF BLOCK b11.
```

```
PARAMETERS:
```

```

p_lang  TYPE sy-langu DEFAULT sy-langu,
ftype   TYPE t7rur0-ftype,
form    TYPE t7rur0-namef,
massn   TYPE massn,

```

```

massg    TYPE massg,
ordnu    TYPE p0298-ordnu,
orddt    TYPE p0298-orddt,
maspr    TYPE char1,
armod    TYPE syarmod,
arver    TYPE p33_arver,
pv_guid  TYPE guid_32.
SELECTION-SCREEN END OF BLOCK b11.

*&-----
-*
*&  Include          Z_BPO03_PRPRG_EVENTS
*&-----*

INITIALIZATION.

START-OF-SELECTION.
PERFORM reinit_sel_data_interval.
PERFORM init_predef_fields.

GET peras.
PERFORM actualize_p0298.
PERFORM add_data_for_predef_fields.
PERFORM get_peras.

END-OF-SELECTION.
PERFORM fill_predef_fields CHANGING gs_tmpl_data.
PERFORM call_form.

*&-----
-*
*&  Include          Z_BPOCOMM_PRPRG_FORMS
*&-----*

*&-----*
*&      Form  REINIT_SEL_PERAMS
*&-----*

FORM reinit_sel_data_interval.
  IF pnpnrr[] IS INITIAL.
    RETURN.
  ENDIF.

TRY.
  DATA(lt_p0298) = zcl_bhrcomm_static=>get_single_pernr_it0298_tab(
    EXPORTING

```

```

        iv_pernr = pnppernr[ 1 ]-low
        iv_begda = pnpbegda
        iv_endda = pnpendda ).
    CATCH zcx_bhrcomm. " Виключні подій пакета Z_BHR_COMMON
        RETURN.
    ENDTRY.

    READ TABLE lt_p0298 INTO DATA(ls_p0298)
        WITH KEY massn = massn massg = massg
            ordnu = ordnu orddt = orddt.
    IF sy-subrc <> 0.
        RETURN.
    ENDIF.
    IF ls_p0298-grpor = abap_false.
        RETURN.
    ENDIF.

    DATA lt_pnppernr LIKE pnppernr[].
    CALL FUNCTION 'HR_RU_PERSONS_FROM_ORDER'
        EXPORTING
            p_p0298 = ls_p0298
        IMPORTING
            begda   = pn-begda
            endda   = pn-endda
        TABLES
            pnr     = lt_pnppernr.
    ENDFORM.

    *&-----*
    *&      Form ACTUALIZE_P0298
    *&-----*
    FORM actualize_p0298 .
        DATA(lt_p0298) = p0298[].

        DELETE p0298 WHERE massn <> massn
        OR massg <> massg
        OR ordnu <> ordnu
        OR orddt <> orddt.

        IF p0298[] IS INITIAL.
            p0298[] = lt_p0298.
            DELETE p0298 WHERE massn <> massn
            OR ordnu <> ordnu

```

```

    OR orddt <> orddt.
ENDIF.

rp_provide_from_last p0298 space pn-begda pn-endda.
ENDFORM.

*&-----*
*&      Form  INIT_PREDEF_FIELDS
*&-----*
FORM init_predef_fields.
  CLEAR go_predef_fields.
  go_predef_fields = NEW #( ).
  go_predef_fields->set_pref_pernr( iv_pref_pernr = pnppernr-low ).
ENDFORM.

*&-----*
*&      Form  add_data_for_predef_fields
*&-----*
FORM add_data_for_predef_fields .
  IF go_predef_fields IS NOT BOUND. RETURN. ENDIF.

  rp_provide_from_last p0001 space p0298-begda p0298-begda.
  rp_provide_from_last p0002 space p0298-begda p0298-begda.

  go_predef_fields->add_pernr_data(
    is_pernr_data = VALUE #( pernr = p0298-
pernr p0001 = p0001 p0002 = p0002 p0298 = p0298 ) ).
ENDFORM.

*&-----*
*&      Form  FILL_PREDEF_FIELDS
*&-----*
FORM fill_predef_fields
  CHANGING cs_tmpl_data TYPE data.

  IF go_predef_fields IS NOT BOUND. RETURN. ENDIF.

  go_predef_fields-
>fill_predef_fields( CHANGING cs_tmpl_data = cs_tmpl_data ).
ENDFORM.

*&-----*
*&      Form  INIT_FORM
*&-----*

```

```

*      text
*-----*
FORM init_form
  CHANGING co_view TYPE REF TO zif_bpocomm_view.

  TRY.
    CLEAR co_view.
    co_view = zcl_bpocomm_static=>get_view_instance( iv_view_class_name = C
ONV #( form ) ).
    "PERFORM set_view_parameters IN PROGRAM uv_set_view_param_prog USING lo
_view IF FOUND.
    "lo_view->show( id_tmpl_data = us_tmpl_data ).
  CATCH cx_static_check INTO DATA(lo_error).
    WHILE lo_error->previous IS BOUND.
      lo_error = CAST #( lo_error->previous ).
    ENDWHILE.
    MESSAGE lo_error->get_text( ) TYPE 'I' DISPLAY LIKE 'E'.
  ENDMETHOD.
ENDFORM.

*&-----*
*&      Form  SHOW_FORM
*&-----*
*      text
*-----*
FORM show_form
  USING co_view TYPE REF TO zif_bpocomm_view
        us_tmpl_data TYPE data.
  IF co_view IS NOT BOUND.
    RETURN.
  ENDIF.

  TRY.
    co_view->show( id_tmpl_data = us_tmpl_data ).
  CATCH cx_static_check INTO DATA(lo_error).
    WHILE lo_error->previous IS BOUND.
      lo_error = CAST #( lo_error->previous ).
    ENDWHILE.
    MESSAGE lo_error->get_text( ) TYPE 'I' DISPLAY LIKE 'E'.
  ENDMETHOD.
ENDFORM.

*&-----*
-*
```

```

*& Include          Z_BP003_PRPRG_FORMS
*&-----*
*&-----*
*&      Form  GET_PERAS
*&-----*
FORM get_peras.
  PERFORM get_from_pernr.
  PERFORM get_from_p0001.
  PERFORM get_from_p0002.
  PERFORM get_from_p0007.
  PERFORM get_from_p0008.
  PERFORM get_from_p0298.
  PERFORM get_from_p9005.
ENDFORM.
*&-----*
*&      Form  CALL_FORM
*&-----*
FORM call_form.
  DATA lo_view TYPE REF TO zif_bpocomm_view.
  PERFORM init_form CHANGING lo_view.
  PERFORM set_form_param USING lo_view.
  PERFORM show_form USING lo_view gs_tmpl_data.
ENDFORM.
*&-----*
*&      Form  SET_VIEW_PARAMETERS
*&-----*
FORM set_form_param
  USING uo_view TYPE REF TO zif_bpocomm_view.
  uo_view->set_parameter(
    EXPORTING iv_param_nam = 'FILE_NAME'
              iv_param_val = |{ p0298-
orddt DATE = ENVIRONMENT }_{ zcl_bhrcomm_static=>get_it0002_sname( p0002 ) }_
{ p0298-ordnu }| ).
ENDFORM.
*&-----*
*&      Form  GET_FROM_PERNR
*&-----*
FORM get_from_pernr .
  gs_tmpl_data-a4 = pernr-pernr.
  gs_tmpl_data-a45 = pernr-pernr.
ENDFORM.

FORM get_from_p0001.

```

```

rp_provide_from_last p0001 space p0298-begda p0298-endda.

*begin -----GET A0 A62-----
DATA lv_edrpo TYPE char8.
CALL FUNCTION 'HR_PAY_UA_GETEDRPOUCODE'
  EXPORTING
    i_bukrs      = p0001-bukrs          " Company Code
  IMPORTING
    e_edrpoucode = lv_edrpo.          " Character field, 8 characters
long
gs_tmpl_data-a62 = lv_edrpo.
*end -----GET A0 A62-----
gs_tmpl_data-a06_1 = |{ p0001-begda DATE = ENVIRONMENT }|.
IF p0001-endda <> '99991231'.
  gs_tmpl_data-a06_2 = |{ TEXT-062 } { p0001-endda DATE = ENVIRONMENT }|.
ENDIF.

*begin -----GET A7 a46_3-----
TRY.
  DATA(ls_p0001_prev) = zcl_bhrcomm_static=>get_single_pernr_it0001(
    iv_pernr = pernr-pernr
    iv_ondat = CONV #( p0298-begda - 1 ) ).

  gs_tmpl_data-a46_3 = gs_tmpl_data-
a7 = zcl_bhrcomm_static=>get_single_plans_ltext(
  EXPORTING
    iv_plans = ls_p0001_prev-plans          " Штатна посада
    iv_ondat = CONV #( p0298-begda - 1 )    " Системна дата
    iv_subty = '9001'                       " Підтип
  ).
CATCH zcx_bhrcomm.
ENDTRY.
*end -----GET A7 a46_3-----

*begin -----GET A8 A47_3-----
TRY.
  gs_tmpl_data-a47_3 = gs_tmpl_data-
a8 = zcl_bhrcomm_static=>get_single_plans_ltext(
  EXPORTING
    iv_plans = p0001-plans          " Штатна посада
    iv_ondat = p0298-begda          " Системна дата
    iv_subty = '9001'               " Підтип
  ).
CATCH zcx_bhrcomm.

```

```

ENDTRY.
*end -----GET A8 A47_3-----

*begin -----GET A10-----
  TRY.
    gs_tmpl_data-a10 = |{ ls_p0001_prev-
persk } { zcl_bhrcomm_static=>get_single_persk_text( iv_persk = ls_p0001_prev
-persk ) }|.
    CATCH zcx_bhrcomm.
  ENDTRY.

*begin -----GET A12-----
  TRY.
    gs_tmpl_data-a12 = |{ p0001-
persk } { zcl_bhrcomm_static=>get_single_persk_text( iv_persk = p0001-
persk ) }|.
    CATCH zcx_bhrcomm.
  ENDTRY.

*begin -----GET A61-----
  CASE ls_p0001_prev-werks.
    WHEN '0101' OR 'NA01'.
      gs_tmpl_data-a61 = TEXT-611.
      gs_tmpl_data-a46_1 = TEXT-611.
    WHEN '0102' OR 'NA02'.
      gs_tmpl_data-a61 = TEXT-612.
      gs_tmpl_data-a46_1 = TEXT-612.
  ENDCASE.

*begin -----GET A61-----
  CASE p0001-werks.
    WHEN '0101' OR 'NA01'.
      gs_tmpl_data-a14 = TEXT-611.
      gs_tmpl_data-a47_1 = TEXT-611.
    WHEN '0102' OR 'NA02'.
      gs_tmpl_data-a14 = TEXT-612.
      gs_tmpl_data-a47_1 = TEXT-612.
  ENDCASE.
*end -----GET A14-----

*begin -----GET A13 a46_2-----
  TRY.
    DATA(lt_rh_struc) = zcl_bhrcomm_static=>get_rh_up_o_struc(

```

```

EXPORTING iv_orgeh = ls_p0001_prev-orgeh iv_ondat = ls_p0001_prev-
begda ).

DATA(lt_rh_object_ltext) = zcl_bhrcomm_static=>get_rh_struc_ltext( lt_r
h_struc ).

SORT lt_rh_object_ltext BY level DESCENDING.

LOOP AT lt_rh_object_ltext ASSIGNING FIELD-
SYMBOL(<ls_rh_object_ltext>).
  IF sy-tabix > 2.
    APPEND VALUE #( a13 = <ls_rh_object_ltext>-
ltext ) TO gs_tmpl_data-tab_a13.

    gs_tmpl_data-a46_2 = |{ gs_tmpl_data-
a46_2 } { <ls_rh_object_ltext>-ltext }|.
    ENDIF.
  ENDLLOOP.
UNASSIGN <ls_rh_object_ltext>.
CATCH zcx_bhrcomm.
ENDTRY.
*end -----GET A13 a46_2-----

*begin -----GET A14_1 a47_2-----
TRY.
  lt_rh_struc = zcl_bhrcomm_static=>get_rh_up_o_struc(
EXPORTING iv_orgeh = p0001-orgeh iv_ondat = p0001-begda ).

  lt_rh_object_ltext = zcl_bhrcomm_static=>get_rh_struc_ltext( lt_rh_stru
c ).

  SORT lt_rh_object_ltext BY level DESCENDING.

  LOOP AT lt_rh_object_ltext ASSIGNING <ls_rh_object_ltext>.
    IF sy-tabix > 2.
      APPEND VALUE #( a14_1 = <ls_rh_object_ltext>-
ltext ) TO gs_tmpl_data-tab_a14_1.

      gs_tmpl_data-a47_2 = |{ gs_tmpl_data-
a47_2 } { <ls_rh_object_ltext>-ltext }|.
      ENDIF.
    ENDLLOOP.
  UNASSIGN <ls_rh_object_ltext>.

```

```

        CATCH zcx_bhrcomm.
    ENDTRY.
*end -----GET A14_1 a47_2-----

*begin -----GET A48-----
    TRY.
        DATA(lv_p1010) = zcl_bhrcomm_static=>get_single_objec_it1010(
    EXPORTING iv_otype = zcl_bhrcomm_static=>mc_otype_s
        iv_objid = CONV #( ls_p0001_prev-plans )
        iv_subty = '9004'
        iv_ondat = p0298-begda ).
        gs_tmpl_data-
a48 = zcl_bhrcomm_static=>get_it1010_hilfm_text( lv_p1010-hilfm ).
        CATCH zcx_bhrcomm.
    ENDTRY.
*end -----GET A48-----

*begin -----GET A49-----
    TRY.
        lv_p1010 = zcl_bhrcomm_static=>get_single_objec_it1010(
    EXPORTING iv_otype = zcl_bhrcomm_static=>mc_otype_s
        iv_objid = CONV #( p0001-plans )
        iv_subty = '9004'
        iv_ondat = p0001-begda ).
        gs_tmpl_data-
a49 = zcl_bhrcomm_static=>get_it1010_hilfm_text( lv_p1010-hilfm ).
        CATCH zcx_bhrcomm.
    ENDTRY.
*end -----GET A49-----
ENDFORM.

*&-----*
*&      Form  GET_FROM_P0298
*&-----*
FORM get_from_p0298 .
*begin -----GET A6-----
    IF p0298-massg = '01'.
        gs_tmpl_data-a6 = TEXT-a61.
    ELSE.
        TRY.
            zcl_bhrcomm_static=>get_single_masss_texts(
                EXPORTING
                    iv_massn      = p0298-massn      " Тип операції

```

```

        iv_massg      = p0298-massg      " Причина операції
        iv_masss      = p0298-
masss " Подобиці причин для кадрових заходів
        iv_ondat      = p0298-begda      " Системна дата
IMPORTING
        ev_mltxt_long = DATA(lv_mltxt_long)      " Подобиці п
по причини кадрових заходів (довгий_текст)
    ).
    CATCH zcx_bhrcomm.
ENDTRY.
gs_tmpl_data-a6 = lv_mltxt_long.
ENDIF.
*end -----GET A6-----

*begin -----GET [A22] A23 A46_4-----
TRY.
    DATA(ls_p0001_prev) = zcl_bhrcomm_static=>get_single_pernr_it0001(
        iv_pernr = pernr-pernr
        iv_ondat = CONV #( p0298-begda - 1 ) ).
    DATA(lt_p1001) = zcl_bhrcomm_static=>get_single_objec_it1001(
EXPORTING iv_otype = zcl_bhrcomm_static=>mc_otype_s
        iv_objid = CONV #( ls_p0001_prev-plans )
        iv_subty = 'B003'
        iv_ondat = p0298-begda ).
    DATA(ls_p1001) = VALUE #( lt_p1001[ sclas = zcl_bhrcomm_static=>mc_otyp
e_a ] DEFAULT VALUE p1001( ) ).
    DATA(ls_p1640) = zcl_bhrcomm_static=>get_single_objec_it1640(
EXPORTING iv_objid = CONV #( ls_p1001-sobid )
        iv_ondat = p0298-begda ).

gs_tmpl_data-a46_4 = gs_tmpl_data-a22 = SWITCH #( ls_p1640-lbcond
    WHEN '0' THEN TEXT-220
    WHEN '1' THEN TEXT-221
    WHEN '2' THEN TEXT-222
    WHEN '3' THEN TEXT-223
    WHEN '4' THEN TEXT-224
    ELSE '' ).
gs_tmpl_data-a23 = |{ ls_p1640-lstnum }, { ls_p1640-zz_lstnum }|.
CATCH zcx_bhrcomm. " Виключні подій пакета Z_BHR_COMMON.
ENDTRY.
*end -----GET [A22] A23 A46_4-----

*begin -----GET [A24] A25-----

```

```

TRY.
  lt_p1001 = zcl_bhrcomm_static=>get_single_objec_it1001(
    EXPORTING iv_otype = zcl_bhrcomm_static=>mc_otype_s
      iv_objid = CONV #( p0001-plans )
      iv_subty = 'B003'
      iv_ondat = p0298-begda ).
  ls_p1001 = VALUE #( lt_p1001[ sclas = zcl_bhrcomm_static=>mc_otype_a ]
    DEFAULT VALUE p1001( ) ).
  ls_p1640 = zcl_bhrcomm_static=>get_single_objec_it1640(
    EXPORTING iv_objid = CONV #( ls_p1001-sobid )
      iv_ondat = p0298-begda ).

  gs_tmpl_data-a24 = SWITCH #( ls_p1640-lbcond
    WHEN '0' THEN TEXT-220
    WHEN '1' THEN TEXT-221
    WHEN '2' THEN TEXT-222
    WHEN '3' THEN TEXT-223
    WHEN '4' THEN TEXT-224
    ELSE '' ).
  gs_tmpl_data-a25 = |{ ls_p1640-lstnum }, { ls_p1640-zz_lstnum }|.
  CATCH zcx_bhrcomm. " Виключні подій пакета Z_BHR_COMMON.
ENDTRY.
*end -----GET [A24] A25-----

gs_tmpl_data-a47_4 = gs_tmpl_data-a24.

*begin -----GET A31-----
TRY.
  gs_tmpl_data-
a31 = zcl_bhrcomm_static=>get_it0298_lgtxt_ltext( iv_lgtxt = p0298-lgtxt ).
  CATCH zcx_bhrcomm.
ENDTRY.
*end -----GET A31-----

*begin -----GET A33 A35 A37 A39 A41 A43-----
TRY.
  DATA(ls_p0002) = zcl_bhrcomm_static=>get_single_pernr_it0002(
    EXPORTING iv_pernr = p0298-chper iv_ondat = p0298-begda ).
  gs_tmpl_data-a33 = zcl_bhrcomm_static=>get_it0002_signam( ls_p0002 ).
  CATCH zcx_bhrcomm.
ENDTRY.

TRY.

```

```

ls_p0002 = zcl_bhrcomm_static=>get_single_pernr_it0002(
    EXPORTING iv_pernr = p0298-zz_chper2 iv_ondat = p0298-begda ).
gs_tmpl_data-a35 = zcl_bhrcomm_static=>get_it0002_signam( ls_p0002 ).
CATCH zcx_bhrcomm.
ENDTRY.

TRY.
ls_p0002 = zcl_bhrcomm_static=>get_single_pernr_it0002(
    EXPORTING iv_pernr = p0298-zz_chper3 iv_ondat = p0298-begda ).
gs_tmpl_data-a37 = zcl_bhrcomm_static=>get_it0002_signam( ls_p0002 ).
CATCH zcx_bhrcomm.
ENDTRY.

TRY.
ls_p0002 = zcl_bhrcomm_static=>get_single_pernr_it0002(
    EXPORTING iv_pernr = p0298-zz_chper4 iv_ondat = p0298-begda ).
gs_tmpl_data-a39 = zcl_bhrcomm_static=>get_it0002_signam( ls_p0002 ).
CATCH zcx_bhrcomm.
ENDTRY.

TRY.
ls_p0002 = zcl_bhrcomm_static=>get_single_pernr_it0002(
    EXPORTING iv_pernr = p0298-zz_chper5 iv_ondat = p0298-begda ).
gs_tmpl_data-a41 = zcl_bhrcomm_static=>get_it0002_signam( ls_p0002 ).
CATCH zcx_bhrcomm.
ENDTRY.

TRY.
ls_p0002 = zcl_bhrcomm_static=>get_single_pernr_it0002(
    EXPORTING iv_pernr = p0298-zz_chper6 iv_ondat = p0298-begda ).
gs_tmpl_data-a43 = zcl_bhrcomm_static=>get_it0002_signam( ls_p0002 ).
CATCH zcx_bhrcomm. " Виключні подій пакета Z_BHR_COMMON
ENDTRY.
*end -----GET A33 A35 A37 A39 A41 A43-----

*begin -----GET [A32] A34 A36 A38 A40-----
TRY.
DATA(ls_p0001) = zcl_bhrcomm_static=>get_single_pernr_it0001(
    EXPORTING iv_pernr = p0298-chper iv_ondat = p0298-begda ).
gs_tmpl_data-a32 = zcl_bhrcomm_static=>get_single_plans_ltext(
    EXPORTING
        iv_plans = ls_p0001-plans           " Штатна посада
        iv_ondat = ls_p0001-begda         " Системна дата

```

```

        iv_subty = '9001' ).
    CATCH zcx_bhrcomm.
ENDTRY.

TRY.
    ls_p0001 = zcl_bhrcomm_static=>get_single_pernr_it0001(
        EXPORTING iv_pernr = p0298-zz_chper2 iv_ondat = p0298-begda ).
    gs_tmpl_data-a34 = zcl_bhrcomm_static=>get_single_plans_ltext(
    EXPORTING
        iv_plans = ls_p0001-plans           " Штатна посада
        iv_ondat = ls_p0001-begda         " Системна дата
        iv_subty = '9001' ).
    CATCH zcx_bhrcomm.
ENDTRY.

TRY.
    ls_p0001 = zcl_bhrcomm_static=>get_single_pernr_it0001(
        EXPORTING iv_pernr = p0298-zz_chper3 iv_ondat = p0298-begda ).
    gs_tmpl_data-a36 = zcl_bhrcomm_static=>get_single_plans_ltext(
    EXPORTING
        iv_plans = ls_p0001-plans           " Штатна посада
        iv_ondat = ls_p0001-begda         " Системна дата
        iv_subty = '9001' ).
    CATCH zcx_bhrcomm.
ENDTRY.

TRY.
    ls_p0001 = zcl_bhrcomm_static=>get_single_pernr_it0001(
        EXPORTING iv_pernr = p0298-zz_chper4 iv_ondat = p0298-begda ).
    gs_tmpl_data-a38 = zcl_bhrcomm_static=>get_single_plans_ltext(
    EXPORTING
        iv_plans = ls_p0001-plans           " Штатна посада
        iv_ondat = ls_p0001-begda         " Системна дата
        iv_subty = '9001' ).
    CATCH zcx_bhrcomm.
ENDTRY.

TRY.
    ls_p0001 = zcl_bhrcomm_static=>get_single_pernr_it0001(
        EXPORTING iv_pernr = p0298-zz_chper5 iv_ondat = p0298-begda ).
    gs_tmpl_data-a40 = zcl_bhrcomm_static=>get_single_plans_ltext(
    EXPORTING
        iv_plans = ls_p0001-plans           " Штатна посада

```

```

        iv_ondat = ls_p0001-begda                " Системна дата
        iv_subty = '9001' ).
    CATCH zcx_bhrcomm.
ENDTRY.
*end -----GET [A32] -----
ENDFORM.

*&-----*
*&      Form  GET_FROM_P0002
*&-----*
FORM get_from_p0002 .
    rp_provide_from_last p0002 space p0298-begda p0298-endda.

    TRY.
        gs_tmpl_data-a5 = gs_tmpl_data-
a44 = zcl_bhrcomm_static=>get_it0002_fname( p0002 ).
        CATCH zcx_bhrcomm.
    ENDTRY.

    TRY.
        gs_tmpl_data-a42 = zcl_bhrcomm_static=>get_it0002_sname( p0002 ).
        CATCH zcx_bhrcomm.
    ENDTRY.
ENDFORM.

*&-----*
*&      Form  GET_FROM_P9005
*&-----*
FORM get_from_p9005 .
    rp_provide_from_last p9005 space p0298-begda p0298-endda.

*begin -----GET A9 A11-----
    TRY.
        DATA(lv_p9005) = zcl_bhrcomm_static=>get_single_pernr_it9005(
            EXPORTING
                iv_pernr = pernr-pernr                " Табельний номер
                iv_ondat = CONV #( p0298-begda - 1 )    " Дата
            ).
        CATCH zcx_bhrcomm.
    ENDTRY.

    TRY.
        gs_tmpl_data-
a9 = zcl_bhrcomm_static=>get_single_it9005_qual_text( lv_p9005-

```

```

qualification ).
    CATCH zcx_bhrcomm.
ENDTRY.
TRY.
    gs_tmpl_data-
all = zcl_bhrcomm_static=>get_single_it9005_qual_text( p9005-qualification ).
    CATCH zcx_bhrcomm.
ENDTRY.
*end -----GET A9 A11-----
ENDFORM.

*&-----*
*&      Form  GET_FROM_P0007
*&-----*
FORM get_from_p0007 .
    rp_provide_from_last p0007 space p0298-begda p0298-endda.

TRY.
    gs_tmpl_data-a26 = zcl_bhrcomm_static=>get_schkz_text(
        EXPORTING is_p0001 = p0001 iv_schkz = p0007-schkz ).
    CATCH zcx_bhrcomm.
ENDTRY.
ENDFORM.

*&-----*
*&      Form  GET_FROM_P0008
*&-----*
FORM get_from_p0008 .
    rp_provide_from_last p0008 space p0298-begda p0298-endda.

TRY.
    gs_tmpl_data-a18 = zcl_bhrcomm_static=>get_it0008_lgart_pbwla(
        EXPORTING is_p0008 = p0008 iv_lgart = '1000' )-betrg.
    CATCH zcx_bhrcomm.
ENDTRY.
*begin -----GET A15-----
TRY.
    DATA(ls_p0008_prev) = zcl_bhrcomm_static=>get_single_pernr_it0008(
        iv_pernr = pernr-pernr
        iv_ondat = CONV #( p0298-begda - 1 ) ).

    gs_tmpl_data-a15 = zcl_bhrcomm_static=>get_it0008_lgart_pbwla(
        EXPORTING is_p0008 = ls_p0008_prev iv_lgart = '1000' )-betrg.

```

```

    CATCH zcx_bhrcomm.
  ENDTRY.
*end -----GET A15-----
*begin -----GET A16 A17 -----
  DATA: lt_lgart_overpay  TYPE RANGE OF lgart,
         lt_lgart_increase TYPE RANGE OF lgart.
  lt_lgart_overpay = VALUE #( sign = 'I' option = 'EQ' ( low = '2030' )
    ( low = '2060' ) ( low = '2070' ) ( low = '2080' ) ( low = '2090' ) ( low =
'2110' )
    ( low = '2150' ) ( low = '2160' ) ( low = '2170' ) ( low = '2290' ) ).
  lt_lgart_increase = VALUE #( sign = 'I' option = 'EQ' ( low = '2500' ) ( lo
w = '2510' )
    ( low = '2520' ) ( low = '2530' ) ( low = '2540' ) ( low = '2550' )
    ( low = '2580' ) ( low = '2590' ) ( low = '2600' ) ( low = '2610' ) ).
  TRY.
    DATA(ls_p0008) = zcl_bhrcomm_static=>get_single_pernr_it0008(
      iv_pernr = pernr-pernr
      iv_ondat = CONV #( p0298-begda - 1 ) ).
    CATCH zcx_bhrcomm.
  ENDTRY.
  TRY.
    DATA(lt_lgart_data) = zcl_bhrcomm_static=>get_it0008_lgarts_pbwla( is_p
0008 = ls_p0008 ).
    IF lt_lgart_data IS NOT INITIAL.
      TRY.
        DATA(lv_molga) = zcl_bhrcomm_static=>get_country_grouping( iv_per
sa = p0001-werks ).
        CATCH zcx_bhrcomm.
      ENDTRY.
      LOOP AT lt_lgart_data ASSIGNING FIELD-SYMBOL(<ls_lgart_data>).
        IF <ls_lgart_data>-lgart IN lt_lgart_overpay.
          APPEND INITIAL LINE TO gs_templ_data-tab_a16 ASSIGNING FIELD-
SYMBOL(<ls_a16_data>).
          TRY.
            <ls_a16_data>-a16 = ||{ CONV i( <ls_lgart_data>-
anzhl ) }{ TEXT-161 } { zcl_bhrcomm_static=>get_single_lgart_ltext(
              iv_lgart = <ls_lgart_data>-
lgart iv_molga = lv_molga ) }|.
            CATCH zcx_bhrcomm.
          ENDTRY.
          ELSEIF <ls_lgart_data>-lgart = '1400'.
            APPEND INITIAL LINE TO gs_templ_data-
tab_a16 ASSIGNING <ls_a16_data>.

```

```

        <ls_a16_data>-a16 = zcl_bhrcomm_static=>get_single_lgart_ltext(
            iv_lgart = <ls_lgart_data>-
lgart iv_molga = lv_molga ).
        UNASSIGN <ls_a16_data>.
    ENDIF.
*-----A17 -----
    IF <ls_lgart_data>-lgart IN lt_lgart_increase.
        APPEND INITIAL LINE TO gs_tmpl_data-tab_a17 ASSIGNING FIELD-
SYMBOL(<ls_a17_data>).
        TRY.
            <ls_a17_data>-a17 = |{ CONV i( <ls_lgart_data>-
anzhl ) }{ TEXT-171 } { zcl_bhrcomm_static=>get_single_lgart_ltext(
                iv_lgart = <ls_lgart_data>-
lgart iv_molga = lv_molga ) }|.
            CATCH zcx_bhrcomm.
        ENDTRY.
        ELSEIF <ls_lgart_data>-lgart = '2570'.
            APPEND INITIAL LINE TO gs_tmpl_data-
tab_a17 ASSIGNING <ls_a17_data> .
            <ls_a17_data>-a17 = |{ <ls_lgart_data>-betrg }{ TEXT-
172 } { zcl_bhrcomm_static=>get_single_lgart_ltext(
                iv_lgart = <ls_lgart_data>-
lgart iv_molga = lv_molga ) }|.
            ENDIF.
        UNASSIGN <ls_a17_data>.
*-----A17 -----
    ENDLOOP.
    UNASSIGN <ls_lgart_data>.
    ENDIF.
    CATCH zcx_bhrcomm.
    ENDTRY.
*end -----GET A16 A17-----
*begin -----GET A18 A19 -----
    TRY.
        lt_lgart_data = zcl_bhrcomm_static=>get_it0008_lgarts_pbwla( is_p0008 =
p0008 ).
        IF lt_lgart_data IS NOT INITIAL.
            LOOP AT lt_lgart_data ASSIGNING <ls_lgart_data>.
                IF <ls_lgart_data>-lgart IN lt_lgart_overpay.
                    APPEND INITIAL LINE TO gs_tmpl_data-tab_a19 ASSIGNING FIELD-
SYMBOL(<ls_a19_data>).
                    TRY.
                        <ls_a19_data>-a19 = |{ CONV i( <ls_lgart_data>-

```

```

anzhl ) }}{ TEXT-191 } { zcl_bhrcomm_static=>get_single_lgart_ltext(
                                iv_lgart = <ls_lgart_data>-
lgart iv_molga = lv_molga ) }|.
    CATCH zcx_bhrcomm.
    ENTRY.
    ELSEIF <ls_lgart_data>-lgart = '1400'.
        APPEND INITIAL LINE TO gs_templ_data-
tab_a19 ASSIGNING <ls_a19_data>.
        <ls_a19_data>-a19 = zcl_bhrcomm_static=>get_single_lgart_ltext(
                                iv_lgart = <ls_lgart_data>-
lgart iv_molga = lv_molga ).
        UNASSIGN <ls_a19_data>.
    ENDIF.
*-----A20 -----
    IF <ls_lgart_data>-lgart IN lt_lgart_increase.
        APPEND INITIAL LINE TO gs_templ_data-tab_a20 ASSIGNING FIELD-
SYMBOL(<ls_a20_data>).
        TRY.
            <ls_a20_data>-a20 = |{ CONV i( <ls_lgart_data>-
anzhl ) }}{ TEXT-201 } { zcl_bhrcomm_static=>get_single_lgart_ltext(
                                iv_lgart = <ls_lgart_data>-
lgart iv_molga = lv_molga ) }|.
            CATCH zcx_bhrcomm.
            ENTRY.
            ELSEIF <ls_lgart_data>-lgart = '2570'.
                APPEND INITIAL LINE TO gs_templ_data-
tab_a20 ASSIGNING <ls_a20_data> .
                <ls_a20_data>-a20 = |{ <ls_lgart_data>-betrg }}{ TEXT-
202 } { zcl_bhrcomm_static=>get_single_lgart_ltext(
                                iv_lgart = <ls_lgart_data>-
lgart iv_molga = lv_molga ) }|.
                ENDIF.
                UNASSIGN <ls_a20_data>.
*-----A20 -----
            ENDLOOP.
            UNASSIGN <ls_lgart_data>.
        ENDIF.
    CATCH zcx_bhrcomm.
    ENTRY.
*end -----GET A19 A20-----
ENDFORM.

```

**ДОДАТОК Б**  
**Результати роботи програми**

ДП НАЕК "Енергоатом"  
 ВП «Запорізька АЕС»  
 Ідентифікаційний код  
 ЄДРПОУ 12345678

### НАКАЗ ПРО ПЕРЕВЕДЕННЯ НА ІНШУ РОБОТУ № 130-КВ від 24.07.2020

Табельний номер 01000017

**Іванов Іван Іванович**

(прізвище, ім'я, по батькові)

Перевести: Тимчасово

з 24.07.2020 по 22.10.2020

Професія, посада	Попередня	ФТ тестова штатна посада			
	Нова				
Розряд, категорія, група	Попередні	2 розряду	Категорія по роду роботи	1С Фахівці	
	Нові	2 розряду		11 Керівники ВЛУ	
Підрозділ, служба, дільниця, лабораторія, відділ, група	Попередні	ПВП ФТ тестова організаційна одиниця			
	Нові	ПВП ФТ тестова організаційна одиниця			
Оклад (тарифна ставка)	Попередні	6898.00 грн.	Доплата		
			Надбавка		
	Нові	6898.00 грн.	Доплата		
			Надбавка		
Форма і система оплати праці	Нові	Почасово-преміальна			
Умови праці	Попередні	ШУ	Список, код	1, 2	
	Нові		виробництва		
Графік роботи Тривалість і режим робочого дня	Новий	40г 7:15 16:30 15:15			
		Початок роботи		Закінчення роботи	
		Перерва			
Індивідуальні умови	Немає				
Підстава для переведення					

Начальник зміни цеху  
атомної електростанції  
(старший)

Микола СІНЦІН

ПОГОДЖЕНО

З наказом  
ознайомлений(а)

Іванов І.І.

Виконавець:  
фахівець УК

## ДОДАТОК ДО НАКАЗУ ПРО ПЕРЕВЕДЕННЯ ПРАЦІВНИКА

Іванов Іван Іванович, таб.№ 0100017				
ПВП ФТ тестова організаційна одиниця ФТ тестова штатна посада ШУ		ПВП ФТ тестова організаційна одиниця		
Не є суб'єктом декларування				
<b>ПІДРОЗДІЛ</b> (попередній, новий)		<b>Відмітки про зняття з обліку та взяття на облік</b>		
<b>Відповідальний</b>		<b>Дата</b>	<b>Посада</b>	<b>Підпис</b>
Табельник попереднього підрозділу				
Табельник нового підрозділу				
Комірник попереднього підрозділу				
Санпропускник				
<b>ВЗПК</b>		<b>Інструктаж із запобігання корупції</b>		
1.Адміністративний корпус каб.№209/01 або каб.№413 2.Інформаційний центр вул. Курчатова 38а (понеділок, середа, п'ятниця з 10-30 до 11-30)		<b>Дата</b>	<b>Підпис особи, яка проводила інструктаж</b>	<b>Прізвище та ініціали особи, яка проводила інструктаж</b>
				<b>Підпис працівника</b>
<b>ПІДРОЗДІЛ</b> (новий)		<b>Відмітка про інструктажі</b>		
<b>З яких питань проінформовано</b>		<b>Дата</b>	<b>Підпис особи, яка проводила інструктаж</b>	<b>Прізвище та ініціали особи, яка проводила інструктаж</b>
Про зміни умов праці на робочому місці				<b>Підпис працівника</b>
Про результати атестації робочого місця				
<b>НТЦ</b>		<b>Психофізіологічне обстеження</b>		
Висновок лабораторії психологічного й психофізіологічного забезпечення УТЦ щодо переводу на оперативні посади, каб. №220 (тел.5-70-00)		..... (Посада) (Підпис) (Ініціали прізвище) (Дата)		
Психофізіологічна експертиза (попередній запис за тел. 5-80-44)		..... (Посада) (Підпис) (Ініціали прізвище) (Дата)		

Психофізіологічне оцінювання	
Висновок щодо психологічного оцінювання професійно-важливих особистих якостей каб. №220 (тел. 5-70-00)	..... (Посада) (Підпис) (Ініціали прізвище) (Дата)
СФЗ	Відмітка про допуск до особливих робіт
Група спецперевірки Прохідна №2 каб. №14 або 17	..... (Посада) (Підпис) (Ініціали прізвище) (Дата)
Бюро перелусток Прохідна №2	..... (Посада) (Підпис) (Ініціали прізвище) (Дата)
Проведено співбесіду щодо переведення	..... (Посада) (Підпис) (Ініціали прізвище) (Дата)

Фахівець (з кадрів) \_\_\_\_\_

Дата \_\_\_\_\_

## Розсилка:

- на паперовому носії: справа, УК (2 прим.);
- в електронному вигляді: бухг., Щось інше, УОНтаОП

Вик. УК

**ДОДАТОК В**  
**Слайди презентації**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

МАГІСТЕРСЬКА РОБОТА НА ТЕМУ:

## **ДОСЛІДЖЕННЯ ТА ПРАКТИЧНЕ ВИКОРИСТАННЯ ХМАРНОЇ ПЛАТФОРМИ SAP ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Виконала:  
ст. гр. КНТ-219М

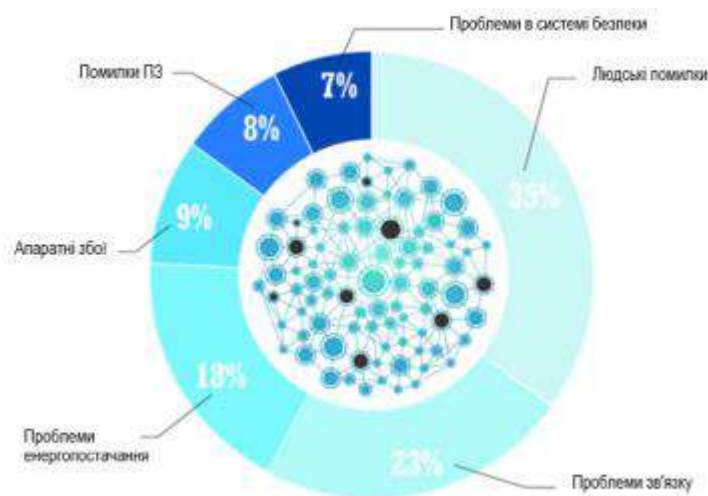
К. С. Полякова

Керівник:  
к.т.н., доц.

А. В. Пархоменко

Рисунок В.1 – Слайд 1

### **АКТУАЛЬНІСТЬ ДОСЛІДЖЕННЯ**



Джерело: [https://www.cisco.com/c/dam/global/ru\\_ru/assets/pdf/mb-04-dna-zk-research\\_10\\_networking\\_priorities-cis-ru.pdf](https://www.cisco.com/c/dam/global/ru_ru/assets/pdf/mb-04-dna-zk-research_10_networking_priorities-cis-ru.pdf)

2

Рисунок В.2 – Слайд 2

## АКТУАЛЬНІСТЬ ДОСЛІДЖЕННЯ



Джерело: <https://gigacloud.ua/en/blog/serzhanko/tema-priznds-iaas-paas-saas>

3

Рисунок В.3 – Слайд 3

## МЕТА ТА ЗАДАЧІ МАГІСТЕРСЬКОЇ РОБОТИ

**Мета роботи:** дослідження та практичне використання хмарних сервісів для підвищення рівня автоматизації документообігу інтелектуального підприємства.

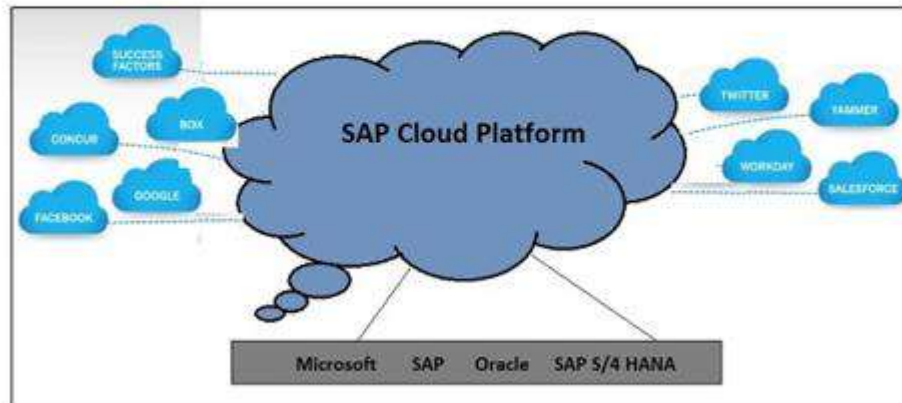
### Задачі роботи:

- дослідити особливості хмарних рішень компанії SAP;
- вивчити методологію DevOps та фреймворк CALMS для підвищення ефективності процесів розробки та експлуатації ПЗ;
- удосконалити методику розробки ПЗ на основі хмарного сервісу з використанням моделі BYOL для мови програмування ABAP/4;
- розробити та впровадити в експлуатацію бізнес - застосунок для автоматичного формування наказів по підприємству для адміністрування персоналу.

4

Рисунок В.4 – Слайд 4

## ДОСЛІДЖЕННЯ АРХІТЕКТУРИ ТА ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ SAP CLOUD PLATFORM

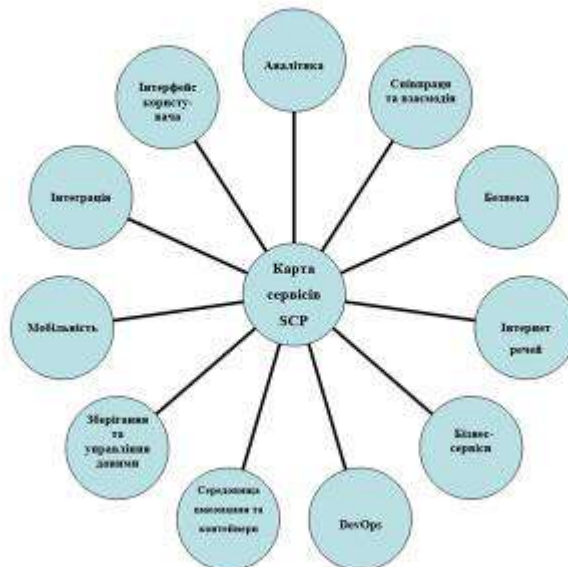


Джерело: <http://codefusions.com/tutorials/sap-architecture-sap-cloud-platform-ibatchais-platform-sap-architecture-ukraine/>

5

Рисунок В.5 – Слайд 5

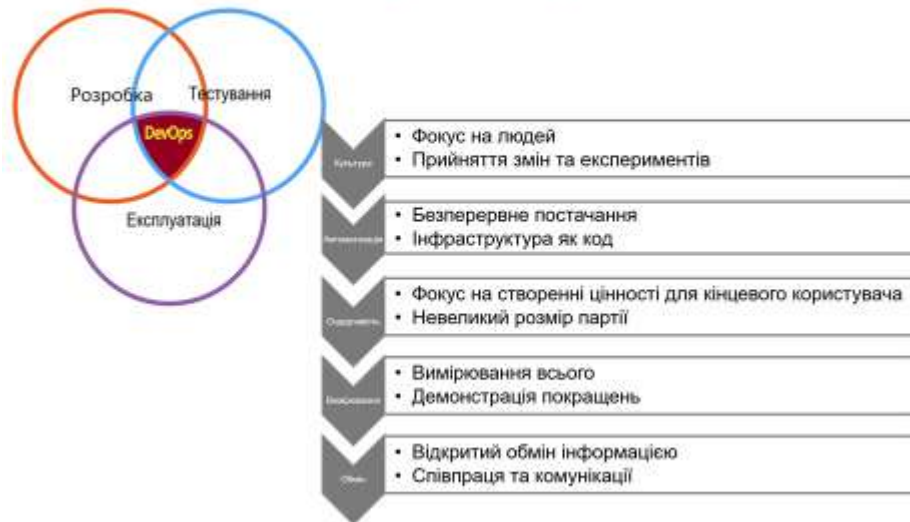
## КАРТА СЕРВІСІВ SAP CLOUD PLATFORM



6

Рисунок В.6 – Слайд 6

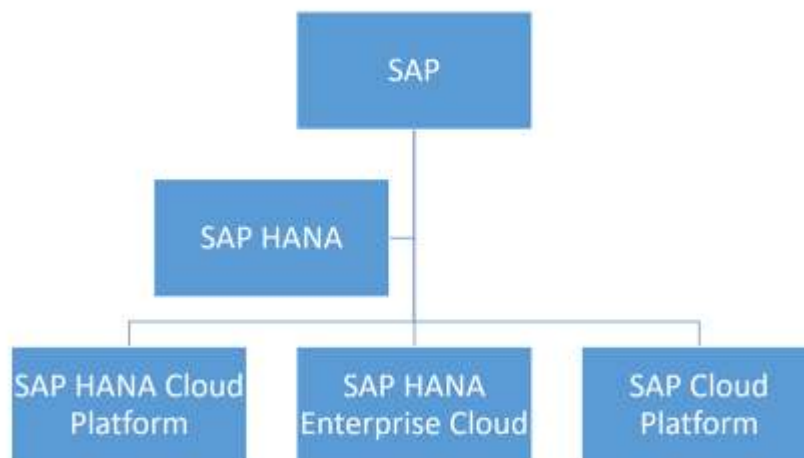
## МЕТОДОЛОГІЯ DEVOPS (DEVELOPMENT & OPERATION) ТА ФРЕЙМВОРК CALMS (CULTURE, AUTOMATION, LEAN, MEASUREMENT, SHARING)



7

Рисунок В.7 – Слайд 7

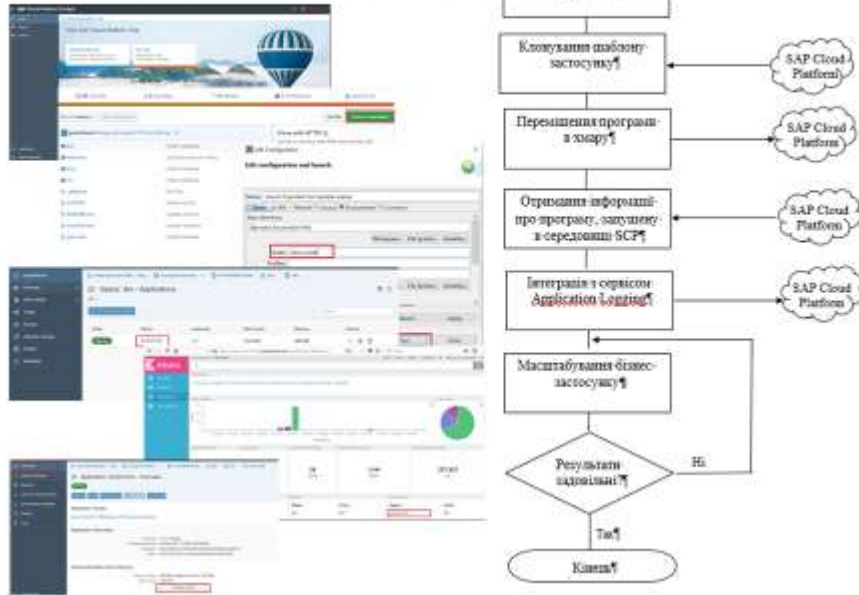
## ВИБІР ХМАРНОГО СЕРВІСУ ТА МОДЕЛІ BYOL (BRING YOUR OWN LANGUAGE)



8

Рисунок В.8 – Слайд 8

## МЕТОДИКА РОЗРОБКИ ХМАРНОГО ЗАСТОСУНКУ



9

Рисунок В.9 – Слайд 9

## УЗАГАЛЬНЕНА СТРУКТУРА БІЗНЕС-ЗАСТОСУНКУ



10

Рисунок В.10 – Слайд 10

## ІНТЕРФЕЙС ЗАСТОСУНКУ

Форма основних даних щодо адміністрування персоналу

Форма перегляду наказів щодо персоналу

Формування відображення наказу та виведення на друк

11

Рисунок В.11 – Слайд 11

## РЕЗУЛЬТАТИ РОБОТИ ЗАСТОСУНКУ

Форма наказу на переведення

Форма додатку до наказу

12

Рисунок В.12 – Слайд 12

## ВИСНОВКИ

В результаті проведених досліджень показано актуальність впровадження хмарних технологій для удосконалення ІТ-ландшафту інтелектуального підприємства, зокрема підвищення рівня автоматизації документообігу.

**Наукове значення** роботи полягає в тому, що удосконалено методику розробки бізнес-застосунку на основі хмарного сервісу, що відрізняється від існуючих підтримкою мови програмування ABAP/4 на основі моделі BYOL.

**Практичне значення** роботи полягає в тому, що розроблений на основі удосконаленої методики бізнес-застосунок дозволяє підвищити рівень автоматизації документообігу підприємства за рахунок зменшення витрат часу та помилок при формуванні наказів на переведення персоналу та внесенні змін у суміжні документи.

13

Рисунок В.13 – Слайд 13

**ДЯКУЮ ЗА УВАГУ!**



Рисунок В.14 – Слайд 14