

Моделювання бізнес процесів

Аналіз вимог до ПЗ доц. Т.І. Каплієнко

Діяльність

- •Повторювана процеси
- •Неповторювана проєкти

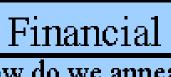


Модель бізнес архітектури

- обізнес процеси / цілі та стратегія побудови бізнесу
- організаційна компонента / організаційне оточення
- оінформація / інформаційне оточення
- озастосунок / оточення, що забезпечує

Аспекти архітектури

- обізнес стратегія
- оархітектура бізнес-процесів
- опоказники результативності



How do we appear to shareholders?

Customer

How do our customers perceive us?

Balanced Scorecard

Internal

At what processes should we excel?

Innovation

What should we learn to grow and prosper?

Balanced Scorecard

Перспектива клієнтів

Яку думку ми маємо створити у наших клієнтів, щоб реалізувати своє бачення майбутнього?

Перспектива фінансів

Яку думку про себе ми маємо створити у наших акціонерів, щоб досягти фінансових успіхів?

Бачення та стратегія

Перспектива навчання та розвитку

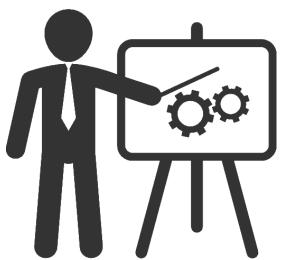
Як ми повинні підтримувати здатність змінюватись і вдосконалюватися, щоб реалізувати своє бачення майбутнього?

Перспектива внутрішній бізнеспроцесів

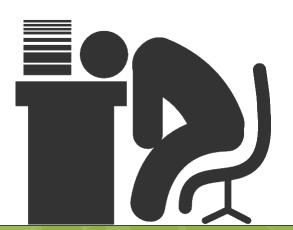
У яких бізнеспроцесах ми маємо досягти досконалості, щоб задовольнити запити наших акціонерів?

Структура організаційної

компоненти









Структура інформаційної компоненти









Единое информационное пространство





Организація компоненти «Застосунки»

Сервер публикации виртуальных рабочих столов Виртуальный Виртуальный рабочий стол рабочий стол Персональный Виртуальный Виртуальный рабочий стол рабочий стол Публикация рабочего стола Hypervisor Социальная сеть Средство мгновенного Веб-почта HTTP/HTTPS обмена сообщениями Эл. почта (SMTP, MAPI) FTP/FTPS Скайп Telnet Карта памяти Принтер Буфер Общие USB обмена сетевые ресурсы (SMB) Облачные файловые хранилища

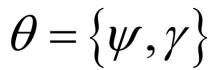
Моделювання

Моделювання (за ISO-15704) – абстрактне уявлення реальності в будь-якій формі (наприклад, в математичної, фізичної, символічної, графічної або дескриптивної), призначене для подання певних аспектів цієї реальності і дозволяє відповідати на питання, які розглядаються.

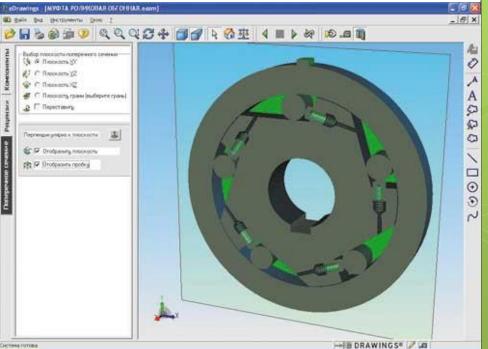


Класифікація моделей

- •Формальні та неформальні
- •Кількісні та якісні
- •Описові та виконувані

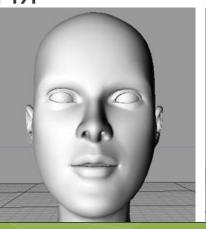


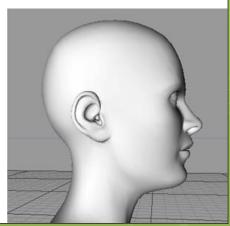




Загальні принципи моделювання

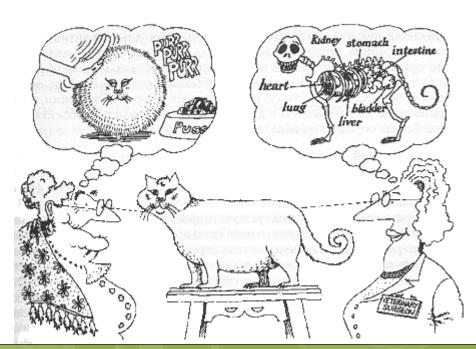
- 1. Принцип здійсненності
- 2. Принцип інформаційної достатності
- 3. Принцип множинності моделі
- 4. Принцип агрегування
- 5. Принцип відділен<u>ня</u>





Об'єктний аналіз

об'єктний аналіз — це метод дослідження не бізнес-процесу в цілому, а його неподільних найменших функціональних частин системи (на даному рівні розгляду) - структурних елементів (об'єктів), пов'язаних між собою деякими відносинами.



Процес



- Це безліч внутрішніх кроків діяльності
- Це потік роботи, що проходить від одного фахівця до іншого
- Це процедура або набір процедур, які спільно реалізують бізнес-завдання
- Це взаємонезалежні компоненти виробничої системи
- Це пов'язаний набір повторюваних дій



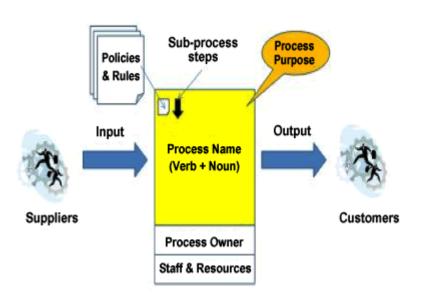


Класифікація бізнес процесів

- Основні процеси
- Процеси, що забезпечують
- Процеси зовнішньої взаємодії

Компоненти процесу

- Назва
- Функція, що реалізується
- Учасники
- Відповідальна особа
- Межі
- Вхідні та вихідні потоки
- Необхідні ресурси
- Мета процесу
- Метрики процесу
- Можливі ризики



Компоненти процесу

• Вхідні потоки

- Ресурси
- Вихідні потоки
- Власник процесу

Enablers

- Resources
- Skills
- Knowledge

Input

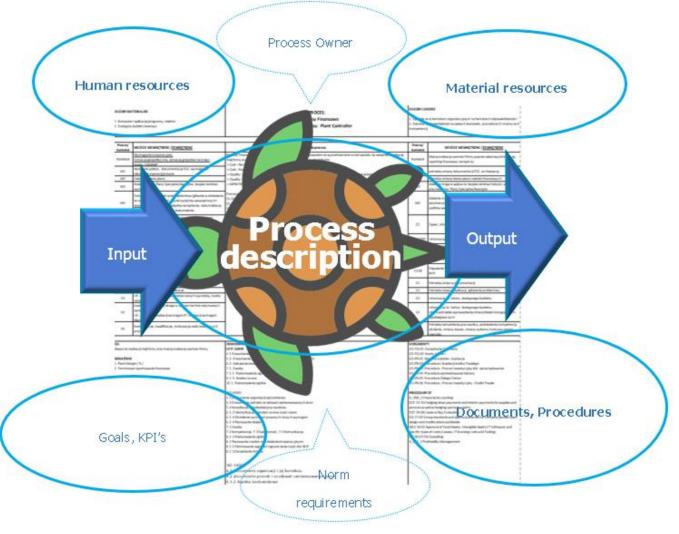
Process

Output

Controls

- Instructions
- Procedures
- Metrics
- Governance

The Turtle Diagram



https://www.automotivequal.com/how-to-describe-processes-in-a-company-using-the-turtle-diagram/

Основні складові моделі бізнес-процесу

- Функції
- Ресурси
- •Документи і дані
- Учасники процесу
- Матеріали / продукти, послуги

Аналіз процесу

- Аналіз топології процесу
- Аналіз характеристик процесу
- Аналіз помилок процесу
- Аналіз динаміки виконання процесу
- Аналіз ризиків процесу
- Аналіз ресурсного оточення процесу
- Аналіз можливостей стандартизації процесу

Характеристики процесу

- •Результативність
- •Визначеність
- Керованість
- Ефективність
- •Повторюваність
- оГнучкість
- •Вартість процесу

Помилки при моделюванні

- Незавершеність
- Невідповідність
- Ієрархічна несумісність
- ((Спадкова)) несумісність



Аналіз динаміки процесів

Імітаційне моделювання - це методика, що дозволяє представляти в рамках динамічної комп'ютерної моделі протікання процесів, дії людей і застосування технологій, що використовуються в досліджуваних процесах.

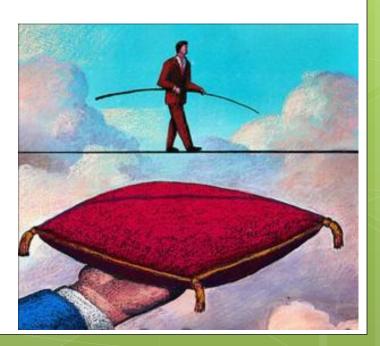




Аналіз ризиків процесу

- 1. Структуризація ризиків
- 2. Опис ризиків та процесів, їх запобігають
- 3. Визначення ризиків в бізнес-процесах

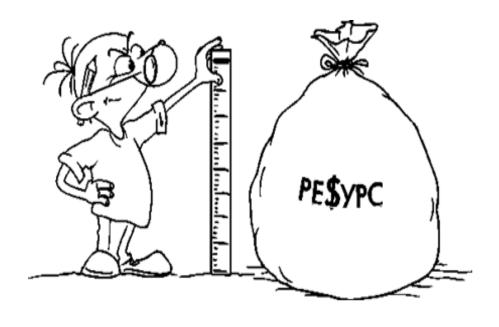




Аналіз ресурсного оточення процесу

Ресурси:

- 1. ЛЮДСЬКІ
- 2. Виробничі
- 3. Матеріальні
- 4. Інформаційні
- 5. Інтелектуальні



Аналіз можливостей стандартизації процесу (створення еталонних, референтних моделей)



Складові моделі об'єкта

- Методики, що використовуються
- Нотація
- Лінгвістичне забезпечення

Види нотацій бізнес процесів

- IDEFO
- BPMN 2.0 (Business Process Modeling Notation)
- o UML ()

BPMN 2.0 - Business Process Model and Notation

http://bpmb.de/poster

End

Activities

Task

A Task is a unit of work, the job to be performed. When marked with a + symbol it indicates a Sub-Process, an activity that can

Transaction

A Transaction is a set of activities that logically belong together; it might follow a specified transaction protocol.

Event Sub-Process An Event Sub-Process is placed into a Process or Sub-Process. It is activated when its start event gets triggered and can interrupt the higher level process context or run in parallel (noninterrupting) depending on the start event.

Call Activity

A Call Activity is a wrapper for a globally defined Task or Process reused in the current Process. A call to a Process is marked with a + symbol.

Activity Markers Markers indicate execution behavior of activities:

+ Sub-Process Marker Loop Marker

Parallel MI Marker

Sequential MI Marker → Ad Hoc Marker

Compensation Marker

Task Types Types specify the nature of the action to be performed:

Send Task Receive Task

User Task Manual Task

Business Rule Task Service Task

Script Task

Sequence Flow

defines the execution

is the default branch to be chosen if all other conditions

evaluate to false

has a condition assigned that defines whether or not the flow is used.

Conditional Flow

Conversations

A Conversation defines a set of logically related message exchanges. When marked with a + symbol it indicates a **Sub-Conversation**, a compound conversation element

A Call Conversation is a wrapper for a globally defined Conversation or Sub-Conversation. A call to a Sub-conversation is marked with a + symbol.

> A Conversation Link connects Conversations and Participants

Conversation Diagram



Choreographies

Participant A Participant A Choreography Task Sub-Choreography + Participant B Participant B A Choreography Task Participant C

(Message Exchange) between two Participants. several Interactions

A Sub-Choreography contains a refined choreography with

A Call Choreography is a wrapper for a globally defined Choreography Task or Sub-Choreography. A call to a Sub-Choreography is marked with a + symbol.

Participant A

Call

Choreography

Participant B

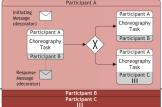
III Participants Marker denotes a set of Participants of the same kind.

Multiple

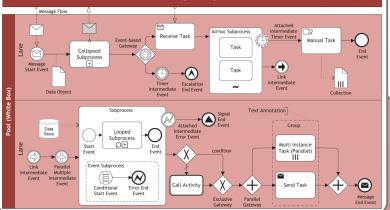
Message a decorator depicting the content of the be attached to Choreography Tasks

Pool (Black Box)

Choreography Diagram



Collaboration Diagram



Events None: Untyped events indicate start point, state changes or final states. Message: Receiving and sending messages. Timer: Cyclic timer events. points in time, time spans or timeouts Escalation: Escalating to an higher level of responsibility. Conditional: Reacting to changed business conditions or integrating business rules. Link: Off-page connectors. (\rightarrow) Two corresponding link events equal a sequence flow. Error: Catching or throwing \otimes named errors. Cancel: Reacting to cancelled (X) \otimes transactions or triggering cancellation. • Compensation: Handling or triggering compensation Signal: Signalling across different processes. A signal thrown can be caught multiple times. Multiple: Catching one out of a set of events. Throwing all events defined Parallel Multiple: Catching all out of a set of parallel Terminate: Triggering the immediate termination of a process.

Gateways



When splitting, it routes the sequence flow to exactly one of the outgoing branches. When merging, it awaits one incoming branch to complete before triggering the outgoing flow. Is always followed by catching events or receive tasks.

Event-based Gateway

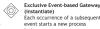
Sequence flow is routed to the subsequent event/task which happens first.

Parallel Gateway

When used to split the sequence flow, all outgoing branches are activated simultaneously. When merging parallel branches it waits for all incoming branches to complete before triggering the outgoing flow.

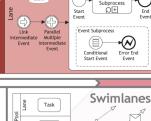
Inclusive Gateway When splitting, one or more branches are activated. All active incoming branches must complete before merging.

Complex Gateway Complex merging and branching behavior that is not captured by other gateways.



event starts a new process instance Parallel Event-based Gateway

(instantiate) The occurrence of all subsequent events starts a new process



Task

or a lane can be an

organization, a role, or a

or other lanes hierarchically.

Message Flow symbolizes information flow across organizational boundaries Pools (Participants) and Lanes

Message flow can be attached represent responsibilities for activities in a process. A pool to pools, activities, or message events. The Message Flow can be decorated with an envelope depicting the system. Lanes subdivide pools content of the message.



→(~)exchanges can be specified by combining RERLIN message flow and sequence flow.









Data

A Data Object represents information flowing through the process, such as business documents, e-mails, or letters.

A Collection Data Object represents a collection of information, e.g., a list of order items.

A Data Input is an external input for the entire process. A kind of input parameter.

A Data Output is data result of the entire process. A kind of output parameter.

A Data Association is used to associate data elements to Activities, Processes and Global

A Data Store is a place where the process can read or write data, e.g., a database or a filing cabinet. It persists beyond the lifetime of the process instance





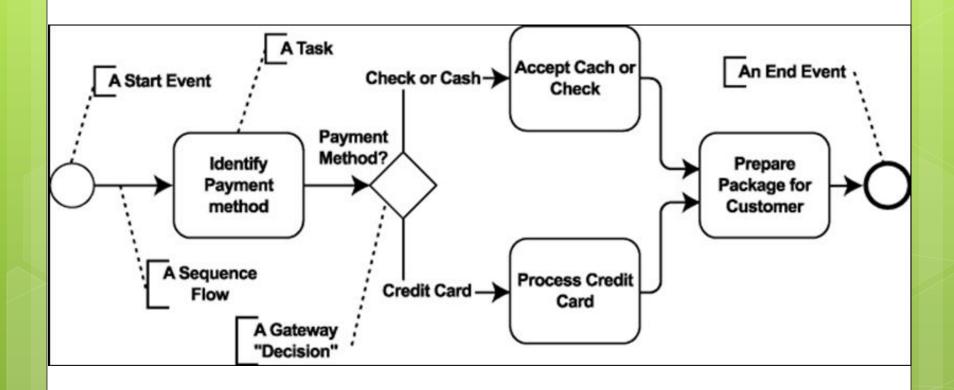
Ш

Input

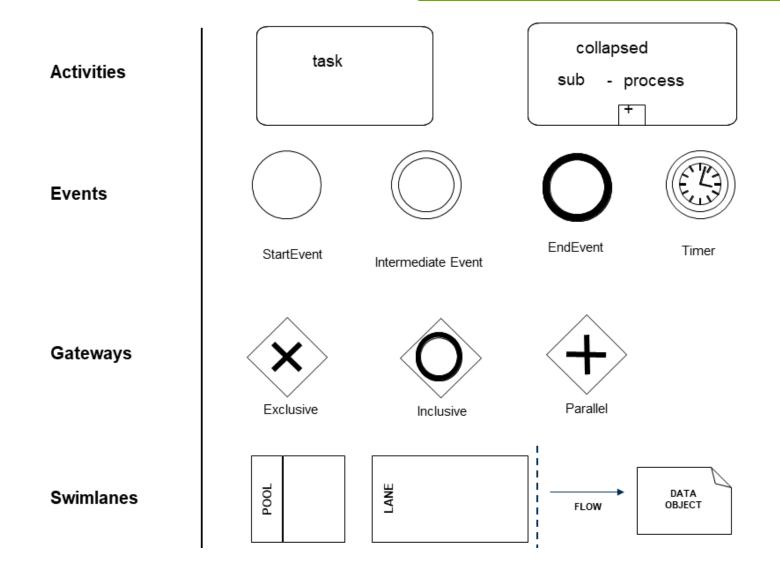
 \rightarrow

Out-

put



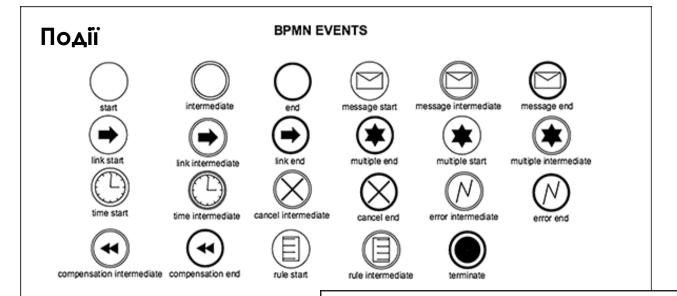
Приклад опису простого процесу в нотації BPMN

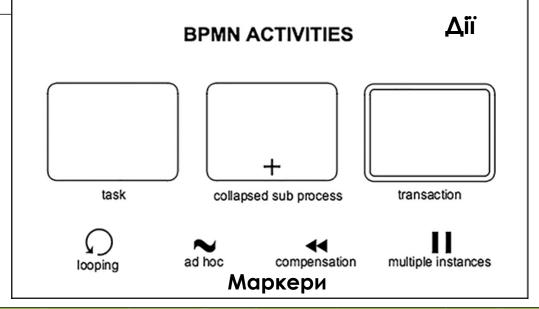


Основні елементи ВРММ

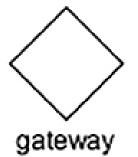
Елементи ВРМN

BPMN 2.0





BPMN GATEWAYS











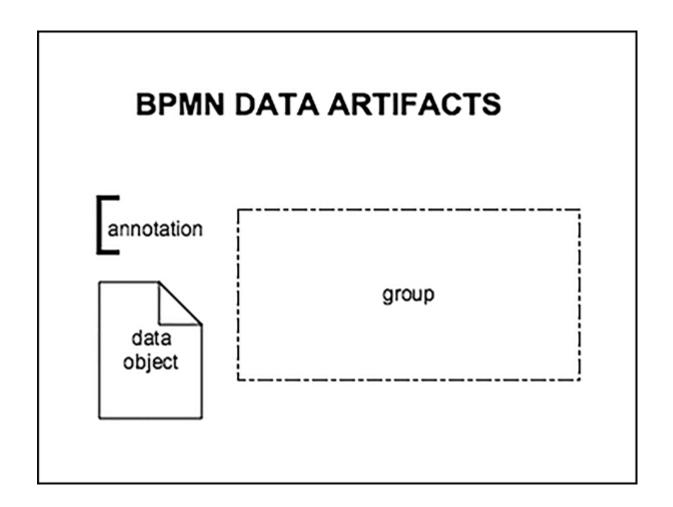


Шлюзи ВРМ N

BPMN CONNECTORS

--- association with arrow --->

Зв'язки ВРММ



Артефакти BPMN

Pools and Lanes



Lane (can be represented vertical)

Name Name

Multiple Lanes (can be represented vertical)



Two Lanes in a Pool (can be represented vertical)



Pool (can be represented vertical)



Pool (with MI Participant) (can be represented vertical)



Pool (with Multiple Lanes) (can be represented vertical)



Collapsed Pool (can be represented vertical)



Collapsed Pool (with MI Participant) (can be represented vertical)

Rows

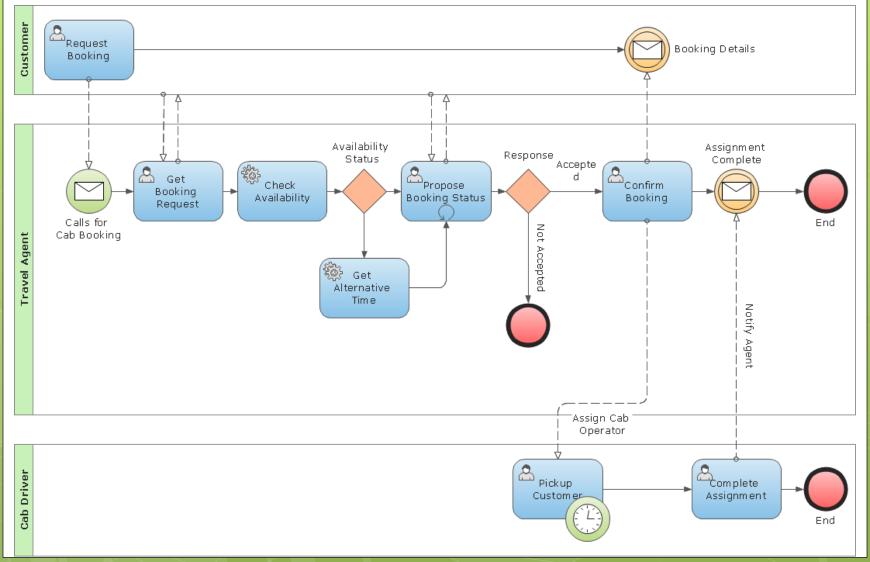




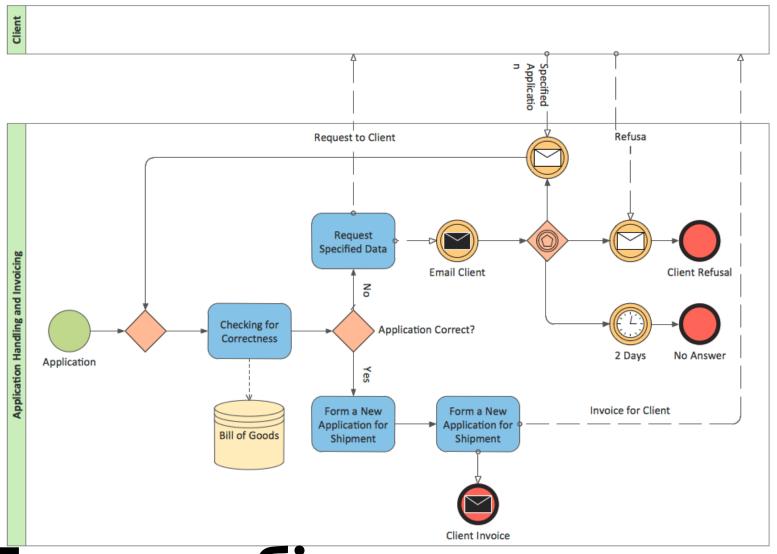
o— — — — → Non-initiating message flow with decorator

Poni BPMN

Діаграма процесу

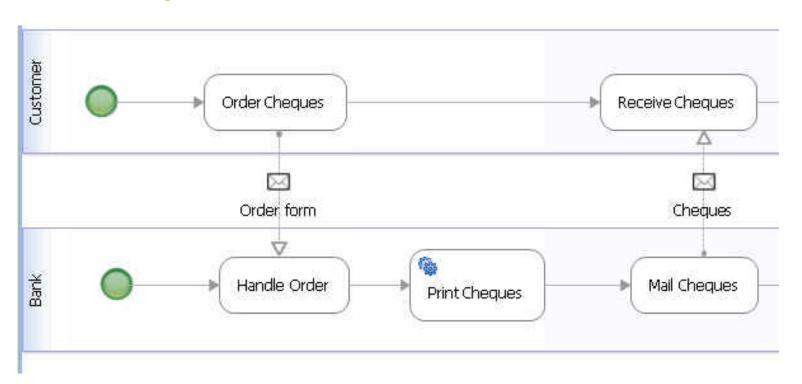


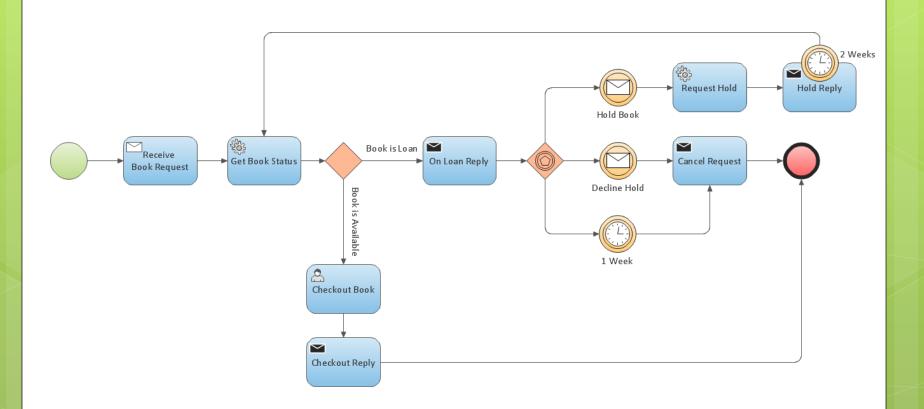
BPMN 2.0

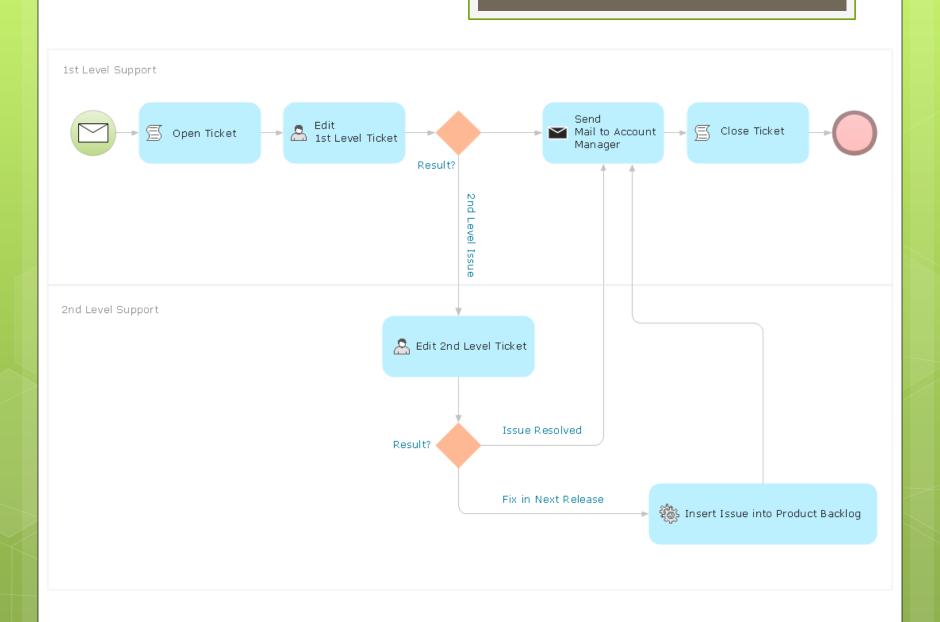


Приклад бізнес- процесу

Діаграма взаємодії

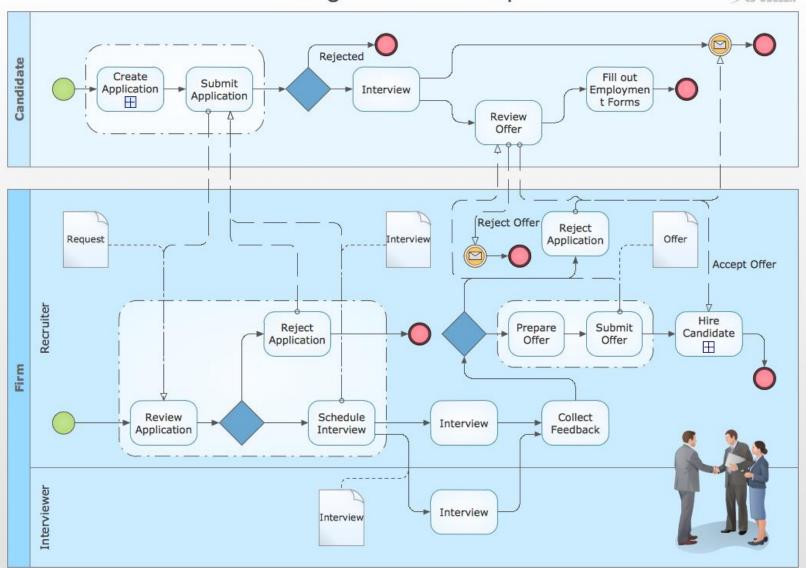






Hiring Process Example





UML

- метод Граді Буча (Grady Booch)
 - Booch 91
 - Booch Light
 - o Booch'93
- Метод Джеймса Румбаха (James Rumbaugh)
 - OMT (Object Modelling Technique)
 - OMT-2
- Метод Айвара Джекобсона (Ivar Jacobson)
 - OOSE (Object-oriented Software Engineering)

UML

- оПредмети
- •Відносини
- •Діаграми

ПРЕДМЕТИ

- структурні предмети;
- предмети поведінки;
- предмети угруповання;
- предмети-коментарі.

1. *Клас* – це опис багатьох об'єктів, що поділяють однакові Властивості, операції, відносини та семантику. Клас реалізує один або декілька інтерфейсів (рис. 2.6).

Клас

Рисунок 2.6 - Клас

Активний клас – це клас чиї об'єкти мають один або декілька процесів (або потоків) і тому можуть ініціювати керуючу діяльність. Активний клас схожий на звичайний клас за винятком того, що його об'єкти діють одночасно з об'єктами інших класів.

2. *Інтерфейс* – набір операцій, що визначають послуги класу або компонента (рис. 2.7). Інтерфейс описує поведінку елемента, видиму ззовні. Інтерфейс може представляти повні послуги класу або компонента. Інтерфейс визначає набір специфікацій операцій, а не набір реалізацій операцій. Інтерфейс рідко показують самостійно. Зазвичай його приєднують до класу або компоненту, який реалізує інтерфейс.

Навчання ○----

Рисунок 2.7 – Інтерфейс

3. *Кооперація* визначає взаємодію і є сукупністю ролей та інших елементів, які працюють разом для забезпечення колективної поведінки більш складного, ніж проста сума всіх елементів. (рис. 2.8) Таким чином, кооперації мають як структурне, так і поведінкове вимірювання. Конкретний клас може брати участь в декількох коопераціях. Ці кооперації являють собою реалізацію патернів, які формують систему.

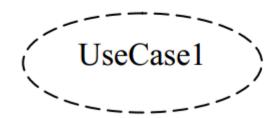


Рисунок 2.8 – Кооперація

4. *Актор* – набір узгоджених ролей, які можуть грати користувачі при взаємодії з системою. Кожна роль вимагає від системи певної поведінки.



Рисунок 2.9 – Актор

5. Елемент *Use-Case* (*Bapiaнт використання* або *Преце- дент*) – це опис послідовності дій (чи декількох послідовностей), що виконуються системою в інтересах окремого актора та формують видимий для актора результат (рис. 2.10). У моделі прецедент застосовується для структурування предметів поведінки. Елемент Прецедент реалізується кооперацією.

UseCase

Рисунок 2.10 – Прецедент

6. **Компонент** — фізична і займана частина системи, що відповідає набору інтерфейсів і забезпечує реалізацію цього набору інтерфейсів (рис. 2.11). У систему включаються як компоненти, які є результатами процесу розробки (файли вихідного коду), так і різні різновиди використовуваних компонентів. Зазвичай компонент — це фізична упаковка різних логічних елементів (класів, інтерфейсів і співробітництв).



Рисунок 2.11 – Компонент

7. **Вузол** – фізичний елемент, що існує в період роботи системи і являє собою деякий ресурс, який зазвичай має пам'ять і можливості обробки (рис. 2.12). У вузлі розміщується набір компонентів, що може переміщатися від одного вузла до іншого.

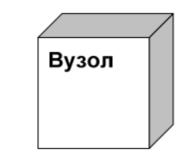


Рисунок 2.12 – Вузол

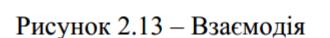
це динамічні частини UML-моделей.

Вони ϵ дієсловами моделей, уявою поведінки в часі і просторі.

Існують дві основні різновиди предметів поведінки.

1. Взаємодія — поведінка, що містить у собі набір повідомлень, якими обмінюється набір об'єктів в конкретному контексті для досягнення певної мети (рис. 2.13).

Взаємодія може визначати динаміку як сукупності об'єктів, так і окремої операції. Елементами взаємодії є повідомлення, які є послідовністю дій та зв'язку.



2. Кінцевий автомат — поведінка, яка визначає послідовність станів об'єкта або взаємодії, що виконуються в ході його існування у відповідь на події (рис. 2.14). З допомогою кінцевого автомата може визначатися поведінка індивідуального класу або кооперації класів. Елементами кінцевого автомата є стани, переходи (від стану до стану), події (предмети, що викликають переходи) та дії (реакції на перехід).

Стан

Рисунок 2.14 – Стан

Ці два елементи взаємодії і кінцеві автомати — ε базисними предметами поведінки, які можуть включатися в *UML* моделі. Семантично ці елементи асоціюються з різними структурними елементами.

Пакет – загальний механізм для розподілу елементів за групами. У пакет можуть поміщатися структурні предмети, предмети поведінки і навіть інші групування предметів. На відміну від компонента, пакет – чисто концептуальне поняття. Це означає, що пакет існує тільки в період розроблення (рис. 2.15).

Інтерфейсні елементи

Рисунок 2.15 – Пакет

Примітка — символ для відображення обмежень і зауважень, приєднуваних до елементу або сукупності елементів.

Примітка

Рисунок 2.16 – Примітка

ВІДНОСИНИ

1 *Залежність* – семантичне відношення між двома предметами, в якому зміна в одному предметі може впливати на семантику іншого предмета (рис. 2.17).

Рисунок 2.17 – Відношення залежності

2. *Асоціація* — структурне відношення, що описує набір зв'язків, які є з'єднанням між об'єктами (рис. 2.18).

-Замовник -Замовлення

Рисунок 2.18 – Асоціація

ВІДНОСИНИ

3. **Узагальнення** – відношення спеціалізації/узагальнення, в якому об'єкти спеціалізованого елемента (нащадка) можуть замінювати узагальнений елемент (предок) (рис. 2.19).

 $\overline{}$

Рисунок 2.19 — Відношення узагальнення

4. *Реалізація* – семантичне відношення між класифікаторами, де класифікатор визначає контракт, який другий класифікатор зобов'язується виконувати (рис. 2.20). Відносини реалізації застосовують у двох випадках: між інтерфейсами і класами (або компонентами), що реалізовують їх; між прецедентами і кооперації, які реалізують їх.

4-----

Рисунок 2.20 – Відношення реалізації

Диаграми

- діаграма класів
- діаграма об'єктів
- діаграма прецедентів
- діаграма послідовності
- діаграма кооперації
- діаграма станів
- діаграма діяльності
- діаграма компонент
- діаграма розміщення

Моделювання поведінки системи засобами UML

- Моделювання прецедентів
- Побудова діаграм діяльності
- Моделювання взаємодій

Опис прецеденту

- Основний виконавець
- Зацікавлені особи
- Передумови
- Післяумови
- Основний успішний сценарій
- Взаємодія між виконавцями
 - верифікація
 - зміна стану системи
 - альтернативні потоки
- Спеціальні вимоги
- Список технологій і типів даних
- Частота використання
- Відкриті питання

Прецедент

- Завершений фрагмент функціональних можливостей і виняткові ситуації
- Фрагмент зовні спостережуваних функцій
- Ортогональний фрагмент функціональних можливостей
- Фрагмент функціональних можливостей, які ініціюються суб'єктом
- Фрагмент функціональних можливостей, які надають суб'єкту корисний результат

