1. The system must enable a client to search products.

2. The system must enable a client to set the number of concurrently displayed items.

3. The system must enable a client to order products.

4. The system must enable a client to cancel an order.

...

99. The system must enable to dispatch an order.

...

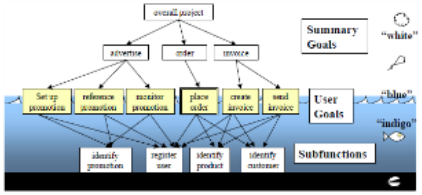
122. It must be possible to add new kinds of products to the system.

...

You have to create a system based on a list of requirements.

What do you find as positive in this (helping you), and what as negative (bothers you)?

What typical interactions with the system – use cases – could you identify in an e-shop?



A. Cockburn. Writing Effective Use Cases. Addison–Wesley, 2000.

Which one of the use cases identified in an e-shop would you realize as the first one?

Writing the typical interactions with the system being created – the use cases – enables to uncover the real intent of the client and to express it comprehensibly, and yet close to code

Use case modularization helps in coping with their complexity (just the include relationship for now)

fiit.sk/~vranic/msoft/dot

UC Place an Order

A customer picks products into the cart. These become a part of their order, which they will eventually confirm, by which the order will move to dispatching.

UC Place an Order

1. Customer selects to place an order.
2. System displays the search options.
3. Customer sets the search options and runs searching.
4. System displays the items that have been found.
5. Customer chooses among the items and confirms s/he choice.
6. System puts the selected items into the cart.
7. Customer can continue in selecting products – the use case continues with step 3.
8. Customer orders the products in the cart.
9. System requests the data necessary to place the order including the payment method.
10. Customer provides the necessary data.
11. Customer can give up the processing of product ordering at any time.
12. System records the order in a list of orders to be dispatched.
13. For each product in the order, System checks the available quantity.
14. If the quantity is below the limit, System adds the quantity under dispatch to the stored plan.
15. The use case ends.

Precondition: Customer is logged on
Protected Data:
- *Historical:* products that have been a part of the order remain there
- *Success:* products ordered by Customer is a part of the order

UC Place an Order

Basic Flow. Main scenario

1. Customer selects to place an order.
2. The items & Products confirm flow is activated.
3. System passes the ordered products into the cart.
4. Customer can continue in selecting products – the use case continues with step 3.
5. Customer orders the products in the cart.
6. System requests the data necessary to place the order including the payment method.
7. Customer provides the necessary data.
8. Customer can group the processes of product ordering at any time.
9. System records the order in a list of orders to be dispatched.
10. For each product in the order, System checks the available quantity.
11. If the quantity is below the limit, System adds the quantity under dispatch to the stored plan.
12. The use case ends.

Alternative flows: Product

1. System displays the search options.
2. Customer sets the search options and runs searching.
3. System displays the items that have been found.

UC Place an Order

Basic Flow. Main scenario

1. Customer selects to place an order.
2. The search options for search products confirm flow is activated.
3. Customer chooses among the items and confirms s/he choice.
4. System puts the selected items into the cart.
5. Customer can continue in selecting products – the use case continues with step 3.
6. Customer orders the products in the cart.
7. System requests the data necessary to place the order including the payment method.
8. Customer provides the necessary data.
9. Customer can give up the processing of product ordering at any time.
10. System records the order in a list of orders to be dispatched.
11. If after dispatching from the stored plan the quantity of any product would fall below the prescribed limit, System adds a record about the needed increment the quantity of the corresponding product to the overall plan.
12. The use case ends.

Alternative flows: Search Products

1. System displays the search options.
2. Customer sets the search options and runs searching.
3. System displays the items that have been found.

Jacobson

```
public class Ordering {
    ...
    public void order(Product product, int quantity) {
        ...
        new SearchProducts().find(product);
        ...
        if (getQuantity(product) >= quantity) {
            ...
        } else...
    }
    ...
}
```

Lecture 1:

Use Cases

Valentino Vranić

Ústav informatiky, informačných systémov
a softvérového inžinierstva

STU FIIT

vranic@stuba.sk fiit.sk/~vranic

MSOFT 2019/20
24. 9. 2019

1. The system must enable a client to search products.

2. The system must enable a client to set the number of concurrntly displayed items.

3. The system must enable a client to order products.

4. The system must enable a client to cancel an order.

...

99. The system must enable to dispatch an order.

...

125. It must be possible to add new kinds of products to the system.

...

1. The system must enable a client to search products.

2. The system must enable a client to set the number of concurrntly displayed items.

3. The system must enable a client to order products.

4. The system must enable a client to cancel an order.

...

99. The system must enable to dispatch an order.

...

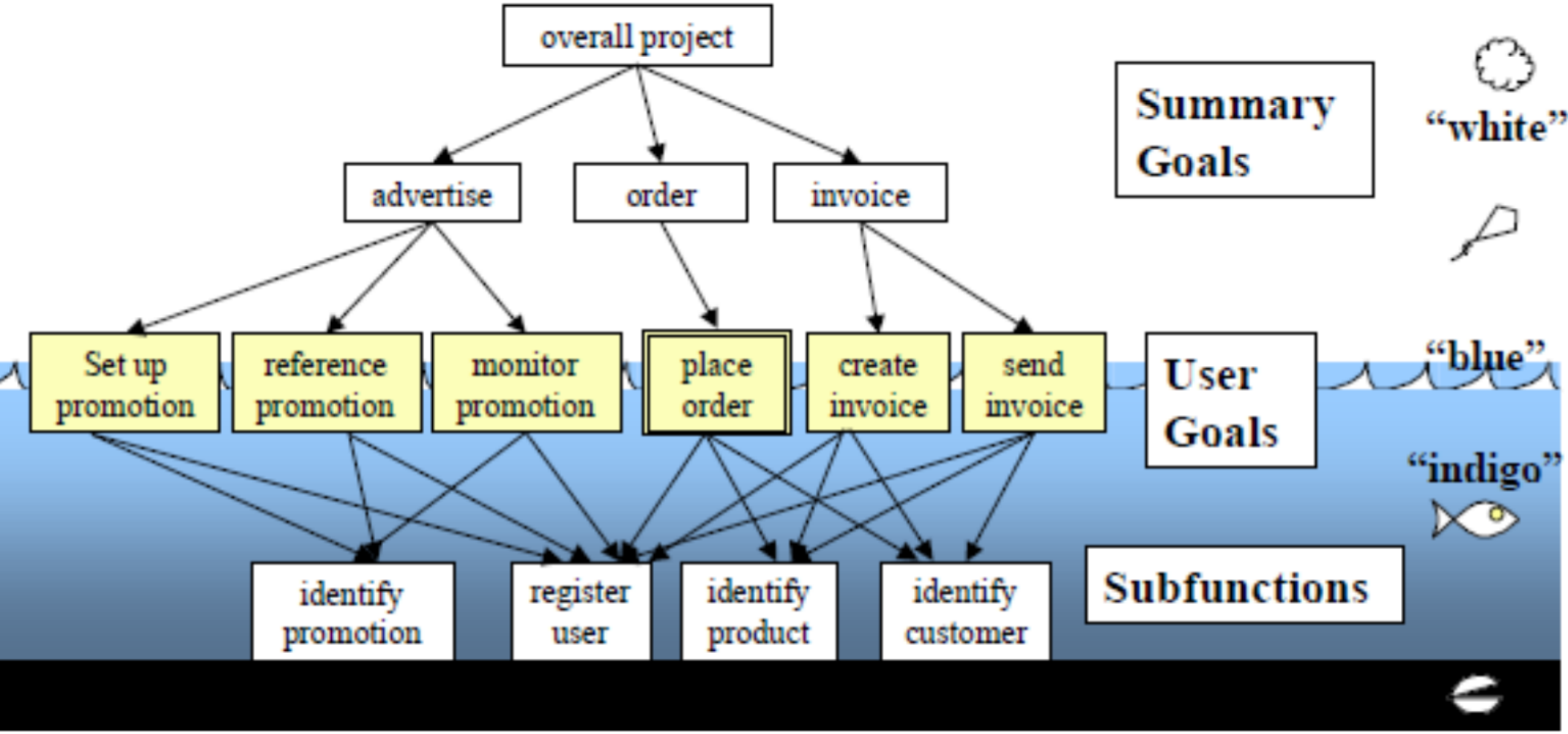
125. It must be possible to add new kinds of products to the system.

...

You have to create a system based on a list of requirements.

What do you find as positive in this (helping you), and what as negative (bothers you)?

What typical interactions with the system
– use cases –
could you identify in an e-shop?



A. Cockburn. Writing Effective Use Cases. Addison–Wesley, 2000.

Which one of the use cases identified in an e-shop would you realize as the first one?

UC Place an Order

A customer picks products into the cart. These become a part of their order, which they will eventually confirm, by which the order will move to dispatching.

UC Place an Order

1. Customer selects to place an order.
2. System displays the search options.
3. Customer sets the search options and runs searching.
4. System displays the items that have been found.
5. Customer chooses among the items and confirms the choice.
6. System puts the selected items into the cart.
7. Customer can continue in selecting products – the use case continues with step 2.
8. Customer orders the products in the cart.
9. System requests the data necessary to place the order including the payment method.
10. Customer provides the necessary data.
11. Customer can give up the processes of product ordering at any time.
12. System records the order in a list of orders to be dispatched.
13. For each product in the order, System checks the available quantity.
14. If the quantity is below the limit, System adds the quantity under demand to the restock plan.
15. The use case ends.

Preconditions: Customer is logged on

Postconditions:

- *Minimal:* products that have been a part of the order remain there
- *Success:* products ordered by Customer is a part of the order

Writing the typical interactions with the system being created – the use cases – enables to uncover the real intent of the client and to express it comprehensibly, and yet close to code

UC Place an Order

1. Customer selects to place an order.
2. System displays the search options.
3. Customer sets the search options and runs searching.
4. System displays the items that have been found.
5. Customer chooses among the items and confirms the choice.
6. System puts the selected items into the cart.
7. Customer can continue in selecting products – the use case continues with step 2.
8. Customer orders the products in the cart.
9. System requests the data necessary to place the order including the payment method.
10. Customer provides the necessary data.
11. Customer can give up the processes of product ordering at any time.
12. System records the order in a list of orders to be dispatched.
13. For each product in the order, System checks the available quantity.
14. If the quantity is below the limit, System adds the quantity under demand to the restock plan.
15. The use case ends.

Preconditions: Customer is logged on
Postconditions:

- *Minimal:* products that have been a part of the order remain there
- *Success:* products ordered by Customer is a part of the order

UC Place an Order

Basic Flow: Place an Order

1. Customer selects to place an order.
2. The Search Products auxiliary flow is activated.
3. System puts the selected products into the cart.
4. Customer can continue in selecting products – the use case continues with step 2.
5. Customer orders the products in the cart.
6. System requests the data necessary to place the order including the payment method.
7. Customer provides the necessary data.
8. Customer can give up the processes of product ordering at any time.
9. System records the order in a list of orders to be dispatched.
10. If after dispatching the order the quantity of any product would fall below the prescribed limit, System adds a record about the need to increase the quantity of the corresponding product to the restock plan.
11. The use case ends.

UC Place an Order

Basic Flow: Place an Order

1. Customer selects to place an order.
2. The Search Products auxiliary flow is activated.
3. System puts the selected products into the cart.
4. Customer can continue in selecting products – the use case continues with step 2.
5. Customer orders the products in the cart.
6. System requests the data necessary to place the order including the payment method.
7. Customer provides the necessary data.
8. Customer can give up the processes of product ordering at any time.
9. System records the order in a list of orders to be dispatched.
10. If after dispatching the order the quantity of any product would fall below the prescribed limit, System adds a record about the need to increase the quantity of the corresponding product to the restock plan.
11. The use case ends.

Auxiliary Flow: Search Products

1. System displays the search options.
2. Customer sets the search options and runs searching.
3. System displays the items that have been found.

Auxiliary Flow: Search Products

1. System displays the search options.
2. Customer sets the search options and runs searching.
3. System displays the items that have been found.

Use case modularization helps in coping with their complexity (just the include relationship for now)

Writing the typical interactions with the system being created – the use cases – enables to uncover the real intent of the client and to express it comprehensibly, and yet close to code

Use case modularization helps in coping with their complexity (just the include relationship for now)

fiit.sk/~vranic/msoft/dot

UC Place an Order

Basic Flow: Place an Order

1. Customer selects to place an order.
2. UC Search Products, its Search Products auxiliary flow, is activated.
3. System puts the selected products into the cart.
4. Customer can continue in selecting products – the use case continues with step 2.
5. Customer orders the products in the cart.
6. System requests the data necessary to place the order including the payment method.
7. Customer provides the necessary data.
8. Customer can give up the processes of product ordering at any time.
9. System records the order in a list of orders to be dispatched.
10. If after dispatching the order the quantity of any product would fall below the prescribed limit, System adds a record about the need to increase the quantity of the corresponding product to the restock plan.
11. The use case ends.

UC Search Products

Auxiliary Flow: Search Products

1. System displays the search options.
2. Customer sets the search options and runs searching.
3. System displays the items that have been found.

Jacobson

UC Place an Order

Basic Flow: Place an Order

1. Customer selects to place an order.
2. UC Search Products, its Search Products auxiliary flow, is activated.
3. System puts the selected products into the cart.
4. Customer can continue in selecting products – the use case continues with step 2.
5. Customer orders the products in the cart.
6. System requests the data necessary to place the order including the payment method.
7. Customer provides the necessary data.
8. Customer can give up the processes of product ordering at any time.
9. System records the order in a list of orders to be dispatched.
10. If after dispatching the order the quantity of any product would fall below the prescribed limit, System adds a record about the need to increase the quantity of the corresponding product to the restock plan.
11. The use case ends.

UC Search Products

Auxiliary Flow: Search Products

1. System displays the search options.
2. Customer sets the search options and runs searching.
3. System displays the items that have been found.

```
public class Ordering {
    ...
    public void order(Product product, int quantity) {
        ...
        new SearchProducts().find(product);
        ...
        if (getQuantity(product) >= quantity) {
            ...
        } else...
    }
    ...
}
```

Jacobson