



ANALYST  
DAYS #10

# Декомпозиция системы

или «ну... пусть это будет подсистема интеграции?»

Шаломович Максим, ЛАНИТ



# Вступление



## Максим Шаломович

- Системный архитектор в компании ЛАНИТ
- Работаю в проектах заказной разработки информационных систем
- Проектирую системы, решения, помогаю технической экспертизой





# Требования к design-time качеству

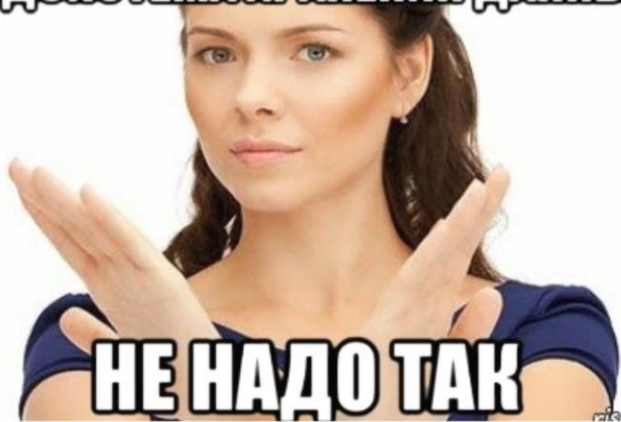


✓ Требования к внутренней структуре

- «Подсистемы системы не должны зависеть друг от друга»
- «Система должна состоять из следующих подсистем»



«Архитектура и ее аналитики» <https://analystdays.ru/ru/talk/68603>



**ПОДСИСТЕМА ХРАНЕНИЯ ДАННЫХ**

**НЕ НАДО ТАК**

**design-time**

**ТВУ**

ANALYST DAYS #9

требования к внутренней структуре

- «Подсистемы системы не должны зависеть друг от друга»
- «Система должна состоять из следующих подсистем»



«Архитектура и ее аналитики» <https://analystdays.ru/ru/talk/68603>



ПОДСИСТЕМА ХРАНЕНИЯ ДАННЫХ



НЕ НАДО ТАК



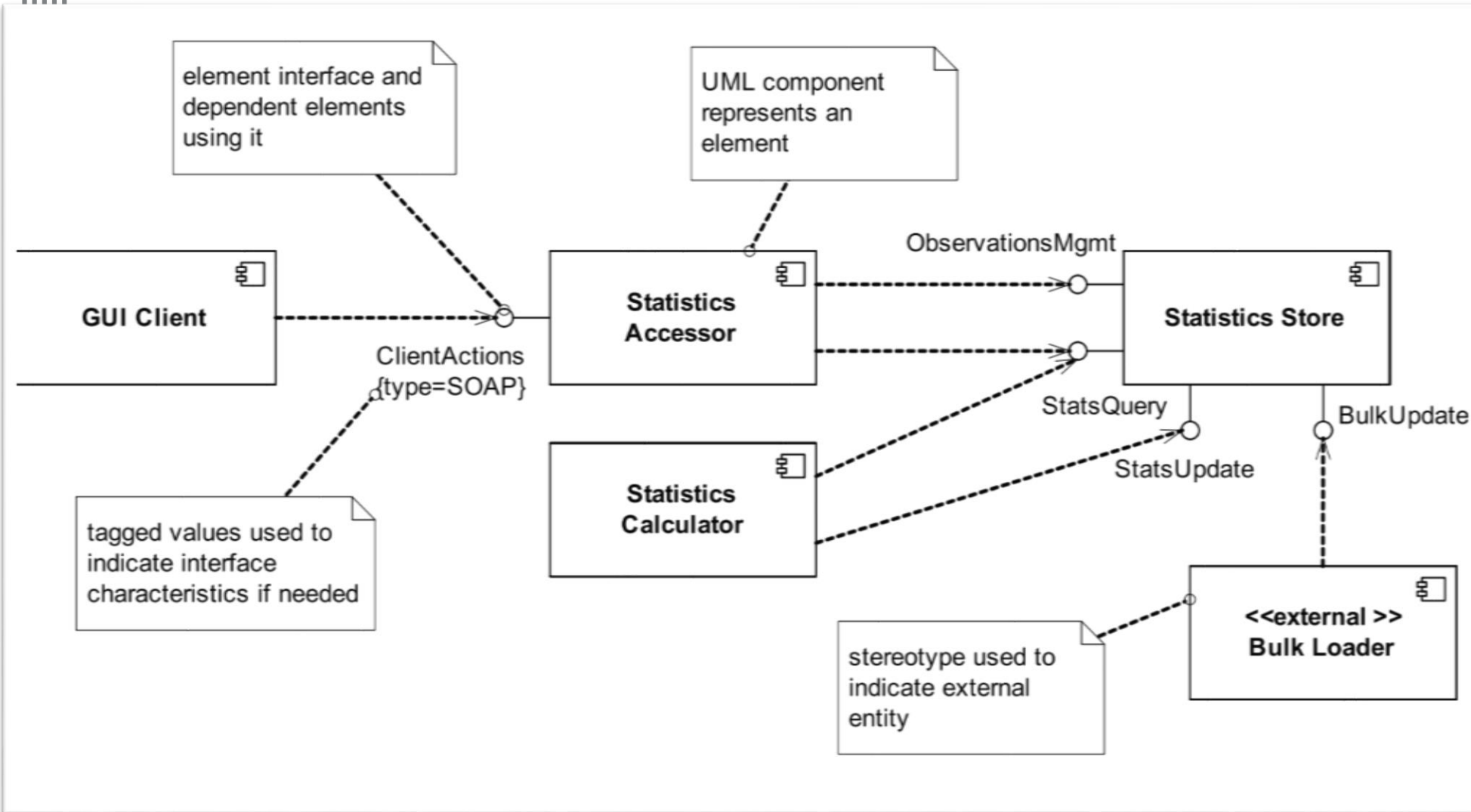
МУАХАХА!

✓ требования к внутренней структуре

- «Подсистемы системы не должны зависеть друг от друга»
- «Система должна состоять из следующих подсистем»



«Архитектура и ее аналитики» <https://analystdays.ru/ru/talk/68603>



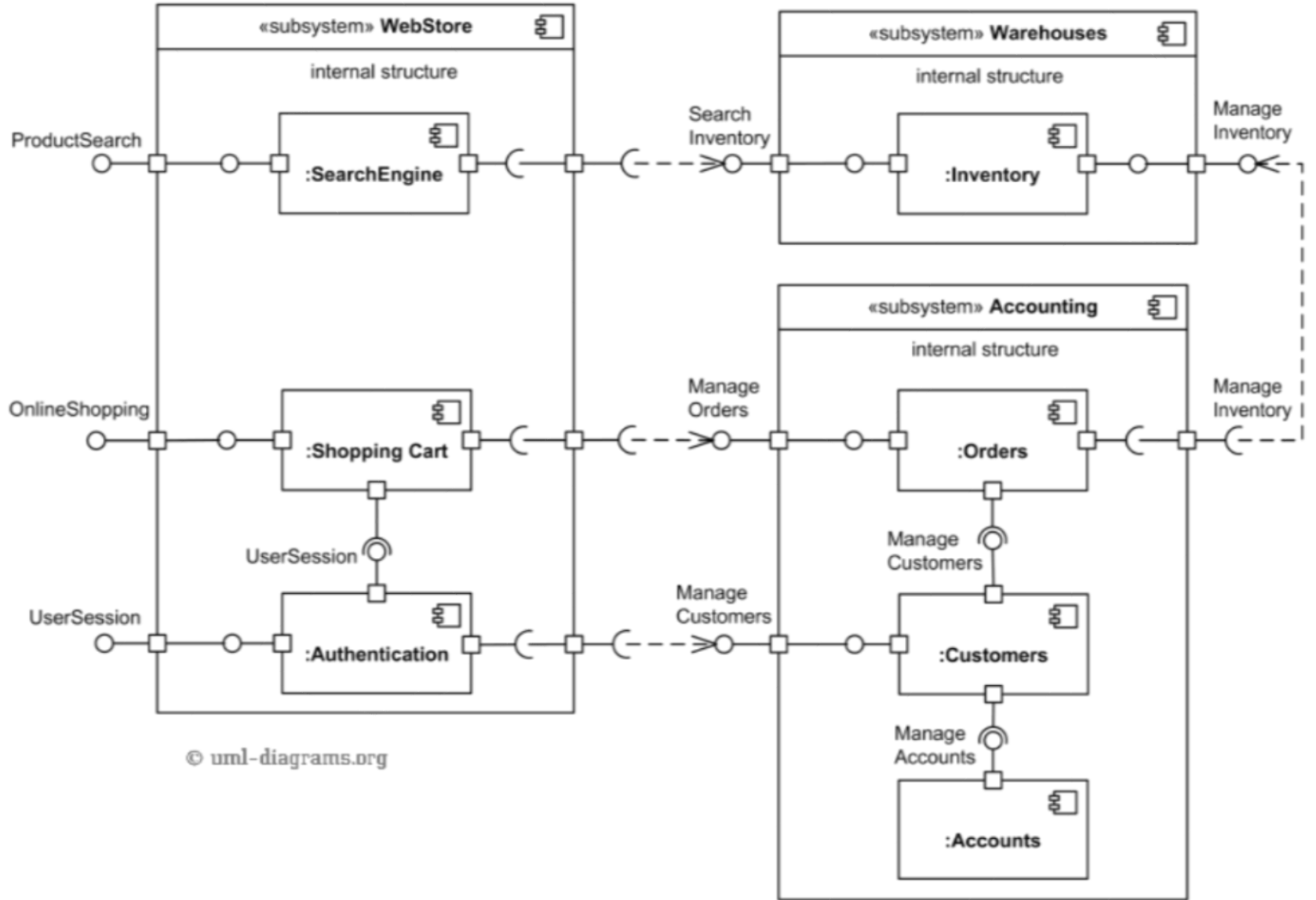


element interface and dependent elements using it



ClientAction:  
{type=SOAP}

tagged values used to indicate interface characteristics if needed





element interface and dependent elements using it

Подсистема

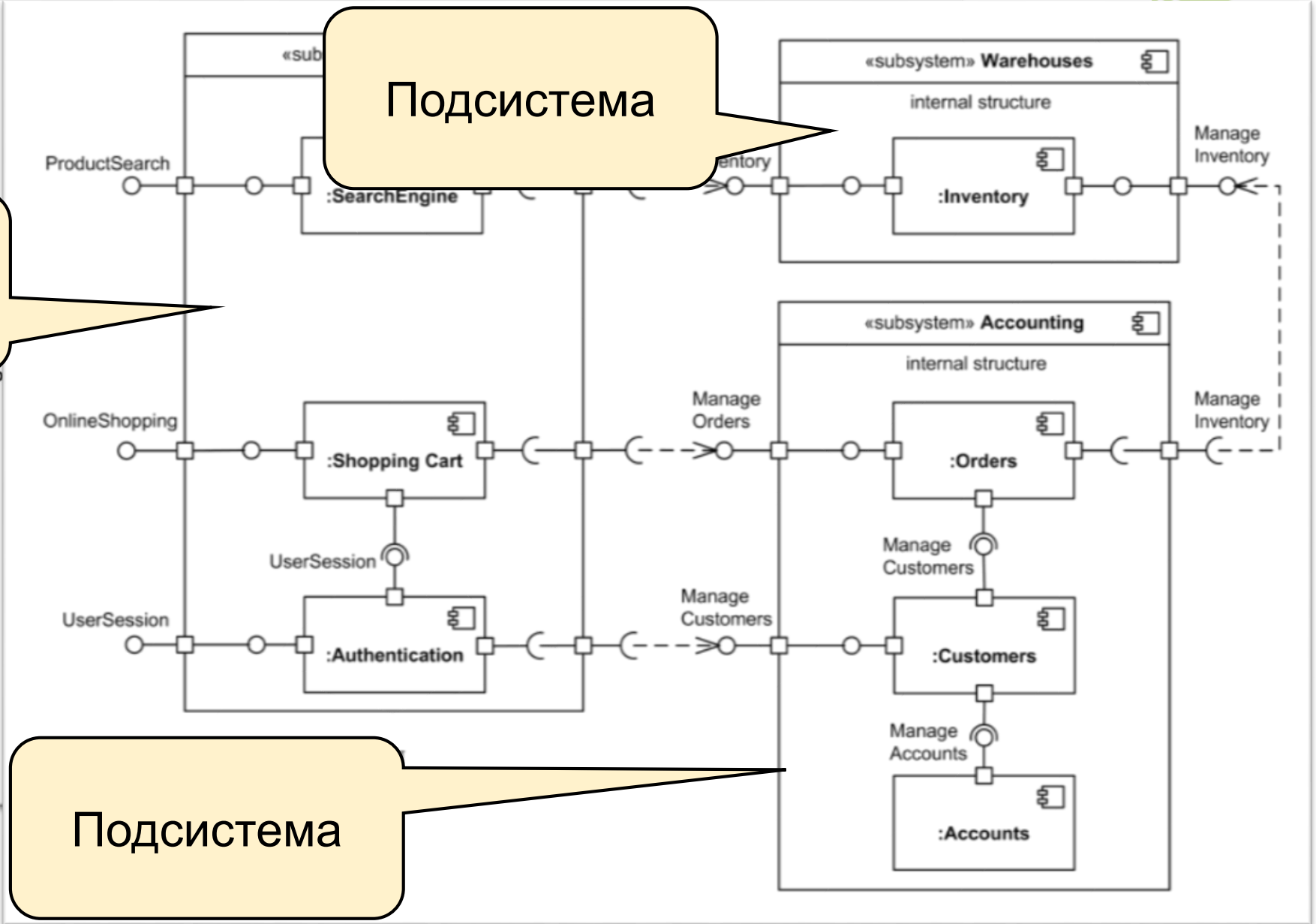
GUI

tagged values used to indicate interface characteristics if needed

type=SOAP

Подсистема

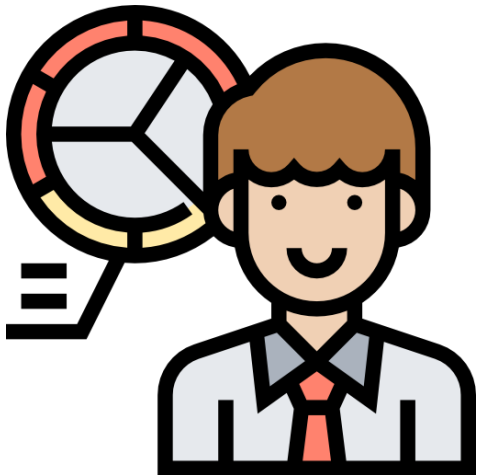
Подсистема





Стадия 0 – Предпроект/ Концептуальное проектирование/....

Стадии 1...n - Анализ и проектирование, разработка, внедрение и т.д.

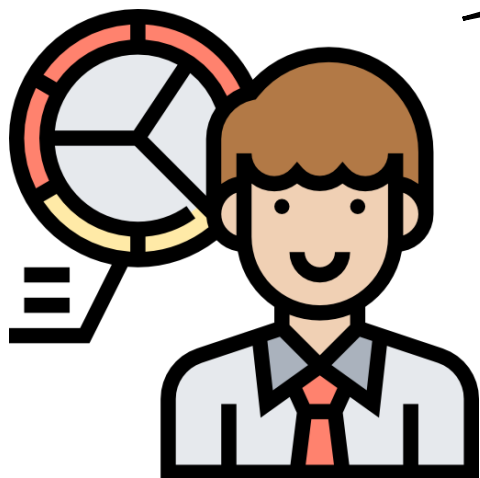




Стадия 0 – Предпроект/ Концептуальное проектирование/....

**Первичная декомпозиция**  
системы (ТЗ, оценка,  
концепция, RFP и т.д.)

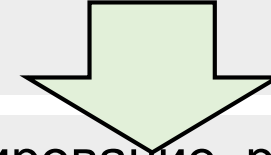
Стадии 1...n - Анализ и проектирование, разработка,  
внедрение и т.д.





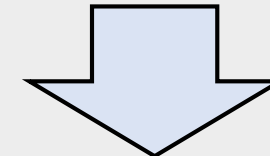
Стадия 0 – Предпроект/ Концептуальное проектирование/....

**Первичная декомпозиция**  
системы (ТЗ, оценка,  
концепция, RFP и т.д.)

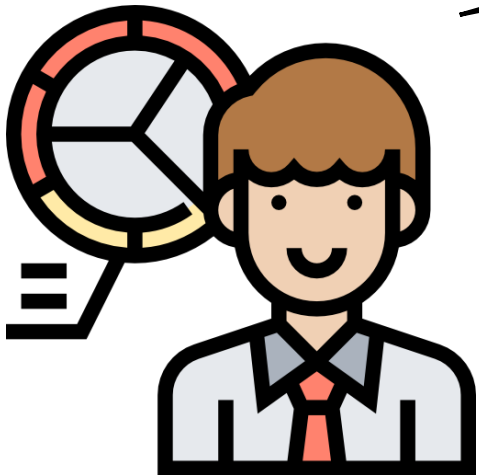


Стадии 1...n - Анализ и проектирование, разработка,  
внедрение и т.д.


Структура работ (по  
«подсистемам»)



Структура артефактов  
(спецификаций, постановок  
задач и т.д.)

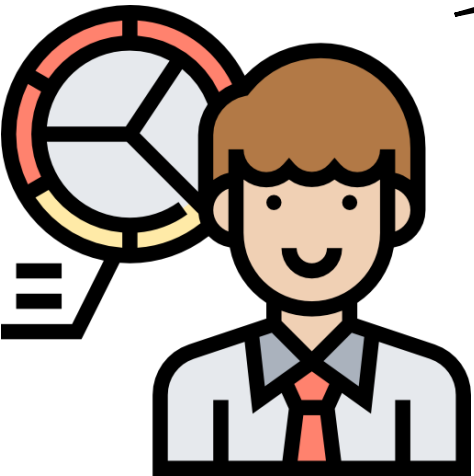


Стадия 0 – Предпроект/ Концептуальное проектирование/....



**Первичная декомпозиция**  
системы (ТЗ, оценка,  
концепция, RFP и т.д.)

Стадии 1...n - Анализ и проектирование, разработка,  
внедрение и т.д.

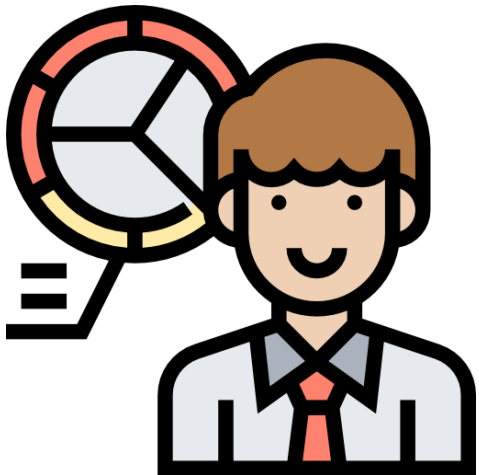


Структура работ (по  
«подсистемам»)

Структура артефактов  
(спецификаций, постановок  
задач и т.д.)



Ща расскажу, зачем и как – и  
декомпозируйте на здоровье!



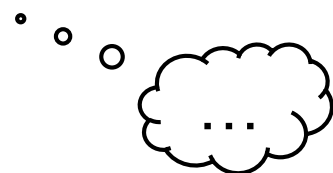
Стадия 0 – Предпроект/ Концептуальное проектирование/....

**Первичная декомпозиция**  
системы (ТЗ, оценка,  
концепция, RFP и т.д.)



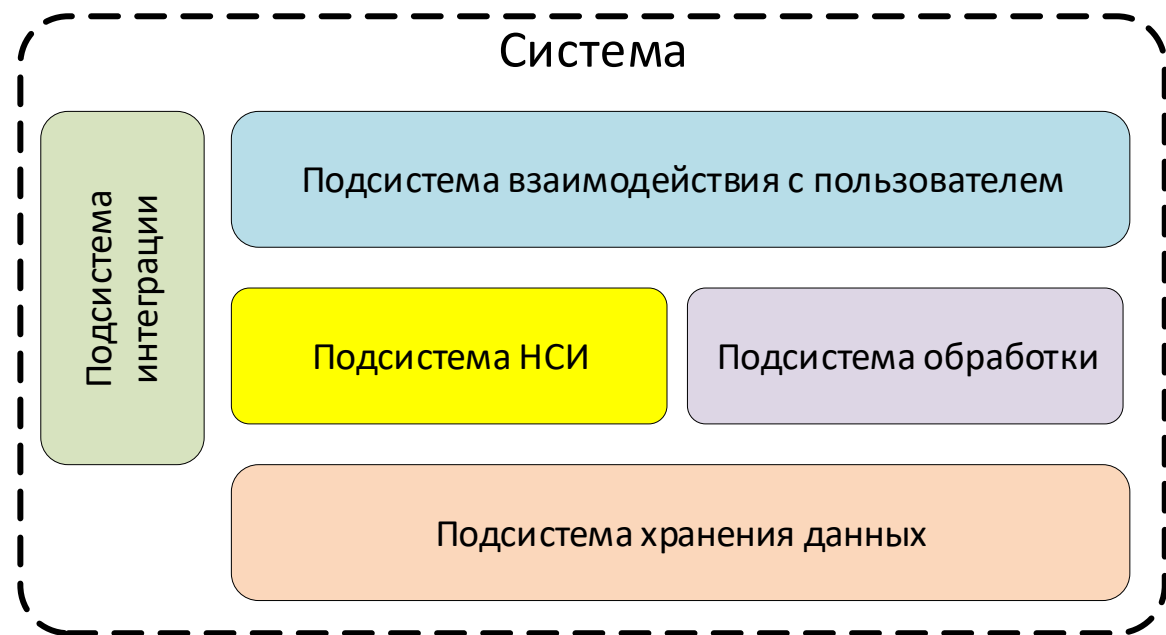
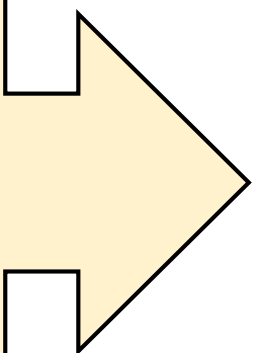
# Критерии декомпозиции

Занудный архитектор вещает



Система предназначена для:

- Хранения данных....
- Обработки данных в соответствии с...
- Предоставления данных в смежные системы по запросу...
- Управления справочниками....
- ...



«Функциональная» декомпозиция

*“On the Criteria To Be Used in Decomposing Systems into Modules”, D.L. Parnas, CMU, 1972*





*“On the Criteria To Be Used in Decomposing Systems into Modules”, D.L. Parnas, CMU, 1972*

Критерии **полезной с точки зрения архитектуры** декомпозиции – она повышает следующие свойства системы:

1. Управляемость (независимые команды)



*“On the Criteria To Be Used in Decomposing Systems into Modules”, D.L. Parnas, CMU, 1972*

Критерии **полезной с точки зрения архитектуры** декомпозиции – она повышает следующие свойства системы:

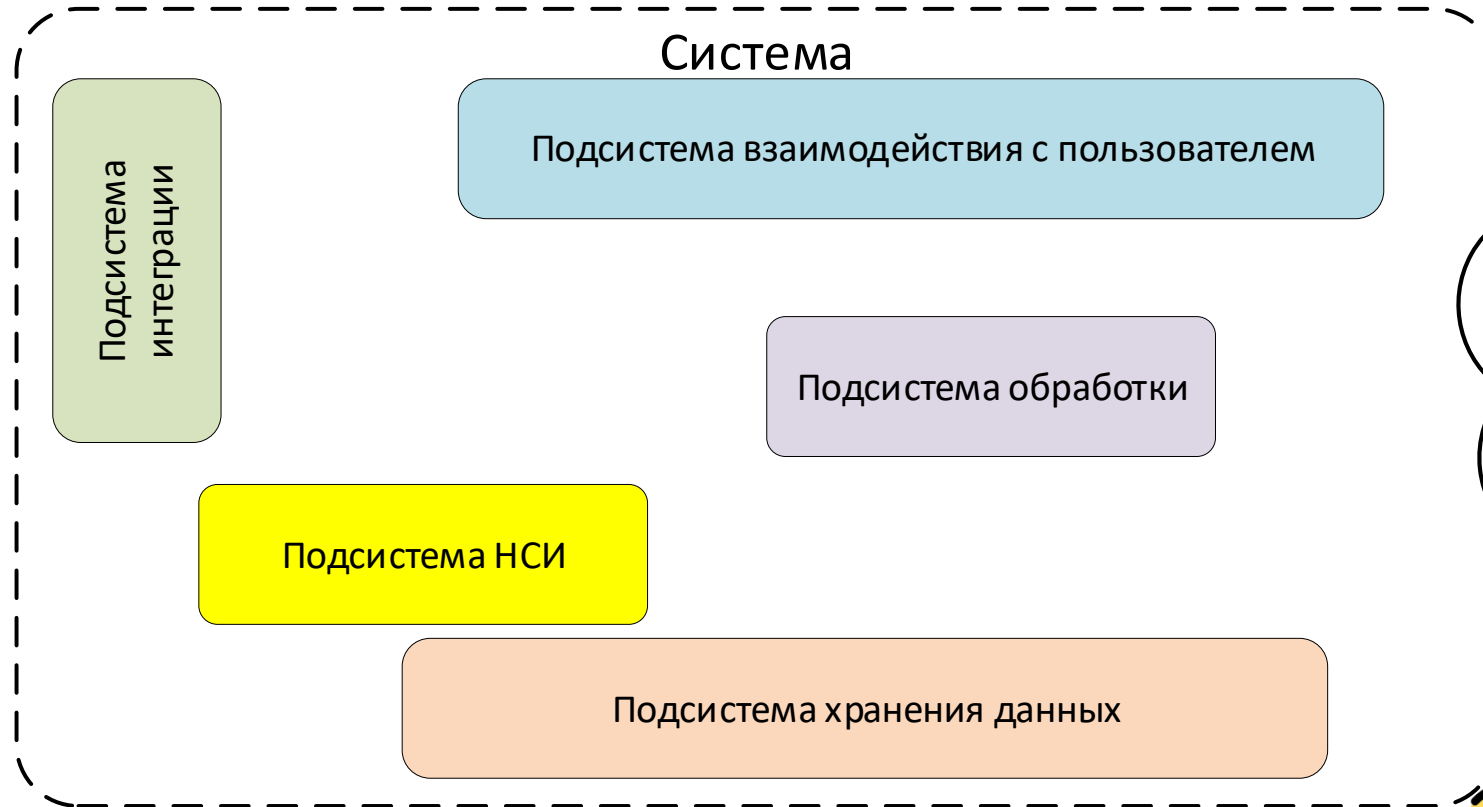
1. Управляемость (независимые команды)
2. Гибкость продукта (внесение изменений)



*“On the Criteria To Be Used in Decomposing Systems into Modules”, D.L. Parnas, CMU, 1972*

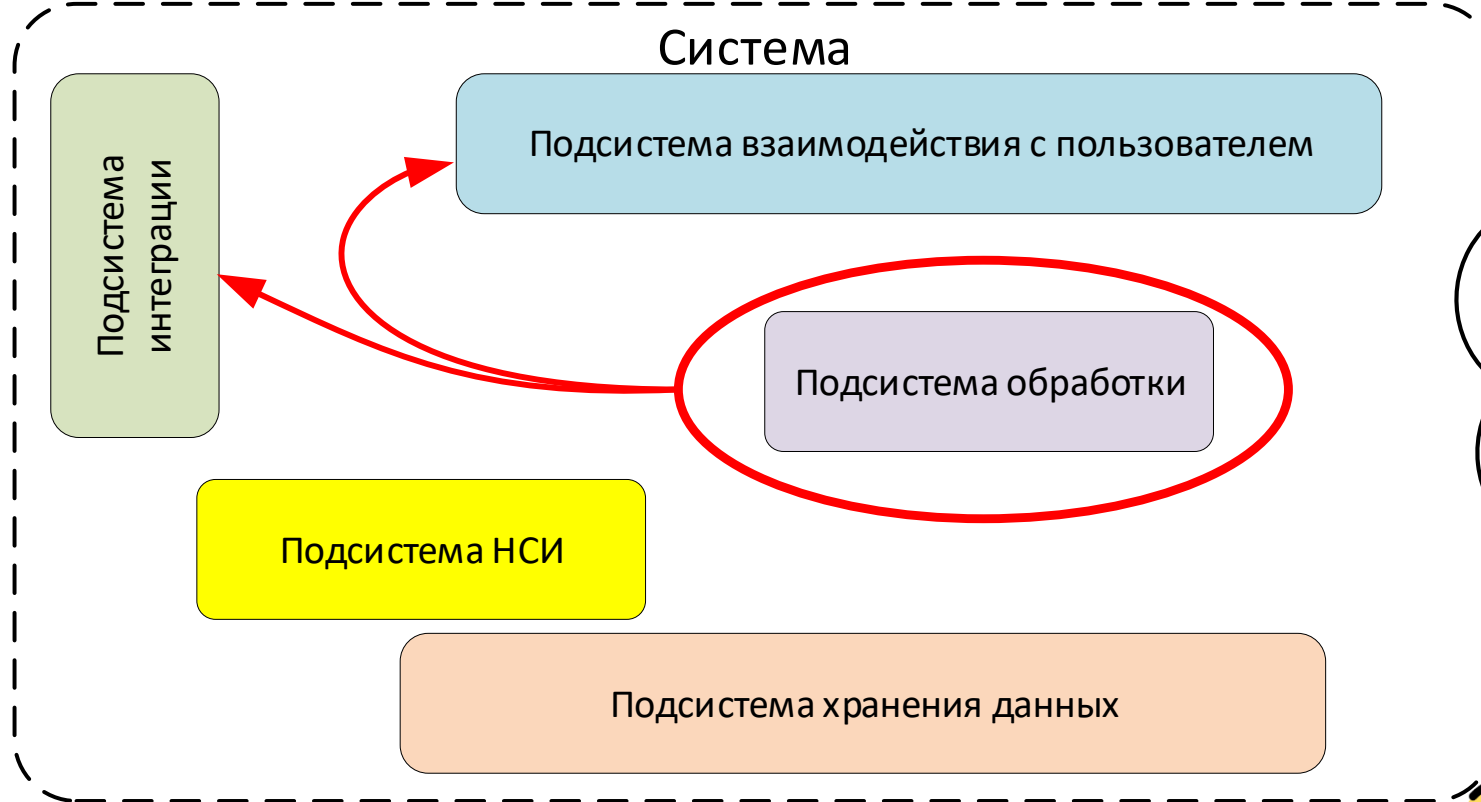
Критерии **полезной с точки зрения архитектуры** декомпозиции – она повышает следующие свойства системы:

1. Управляемость (независимые команды)
2. Гибкость продукта (внесение изменений)
3. «Усвояемость» (возможность изучения системы по частям)



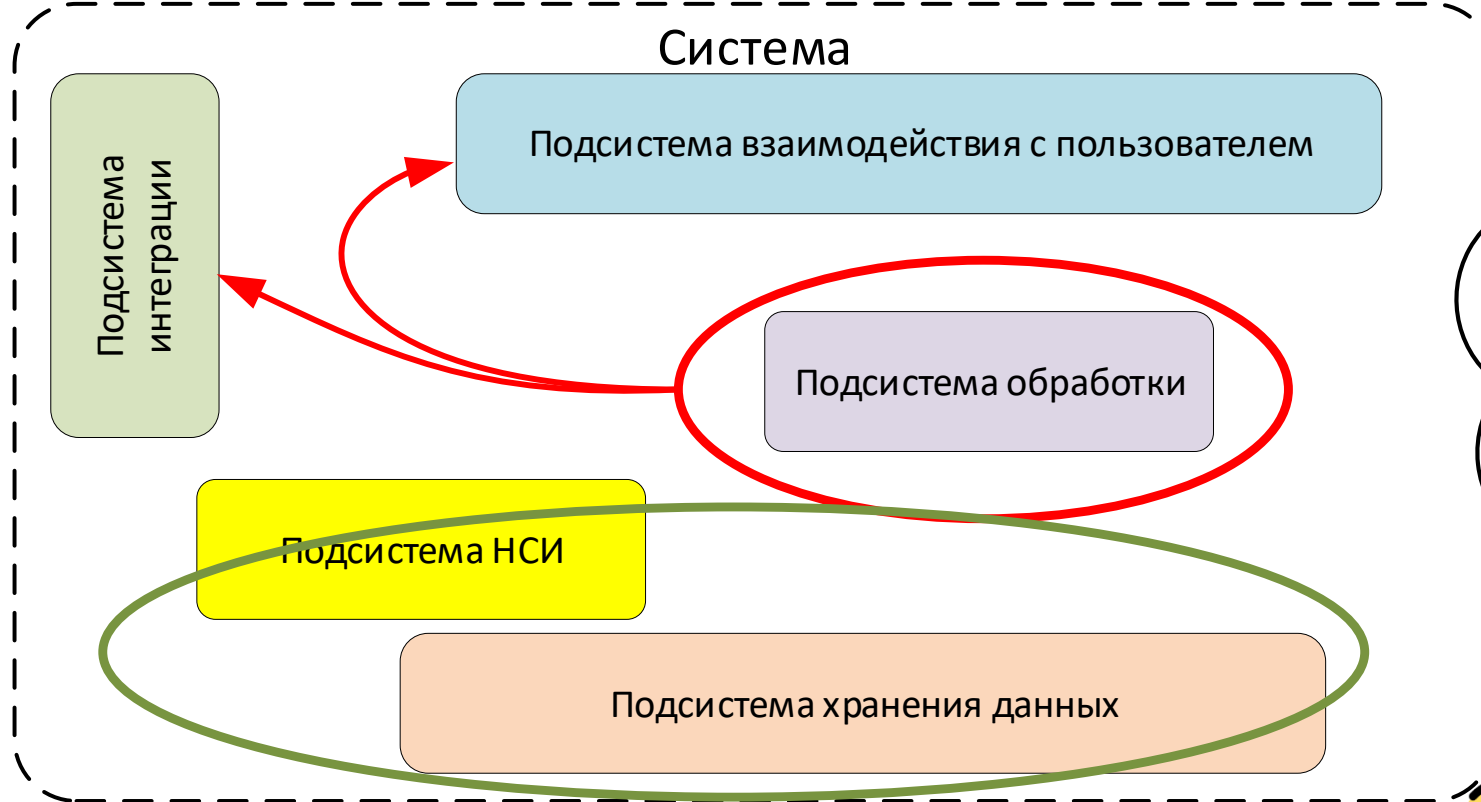
- Управляемость (независимые команды)
- Гибкость продукта (внесение изменений)
- «Усвояемость» (возможность изучения системы по частям)





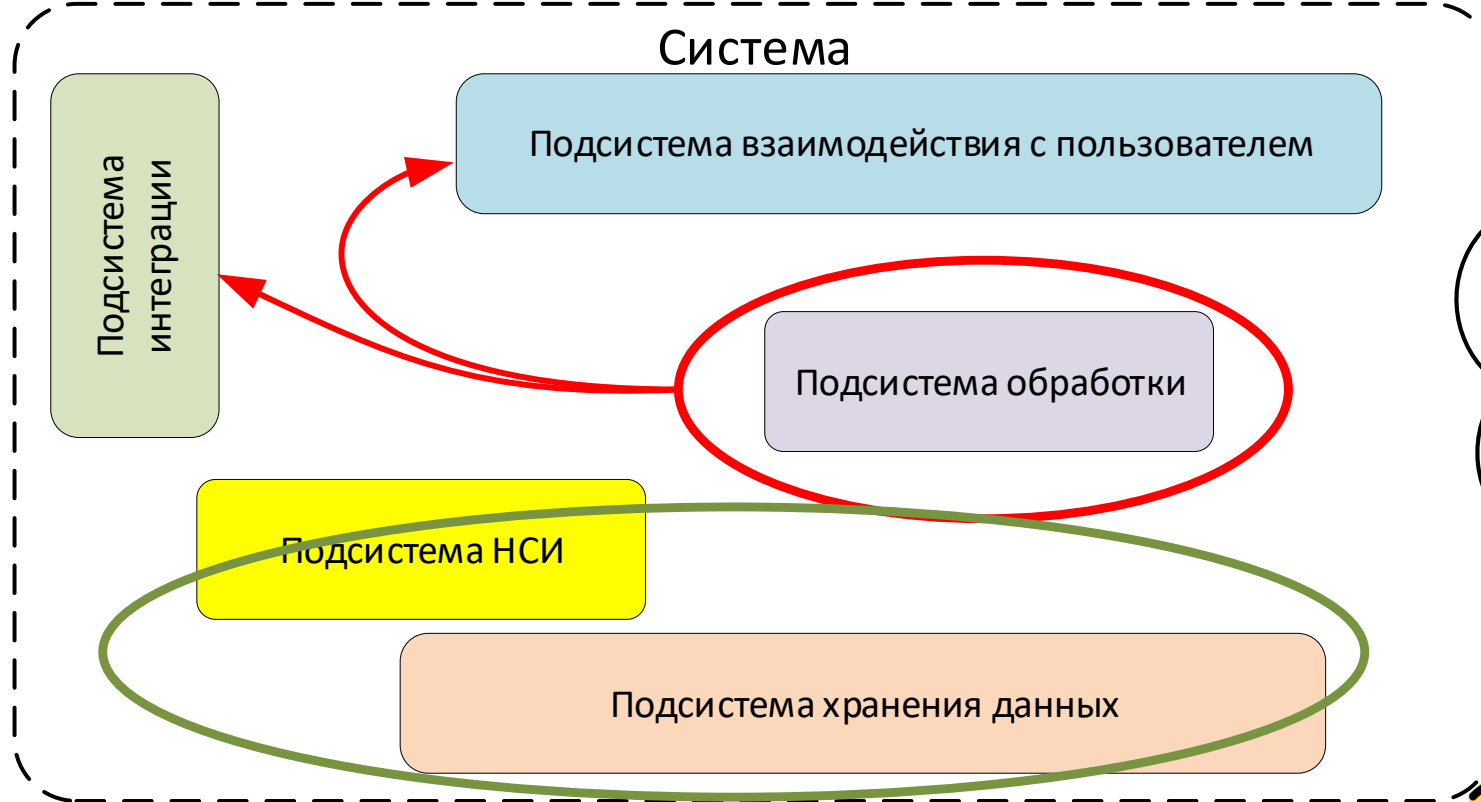
- Управляемость (независимые команды)
- Гибкость продукта (внесение изменений)
- «Усвояемость» (возможность изучения системы по частям)





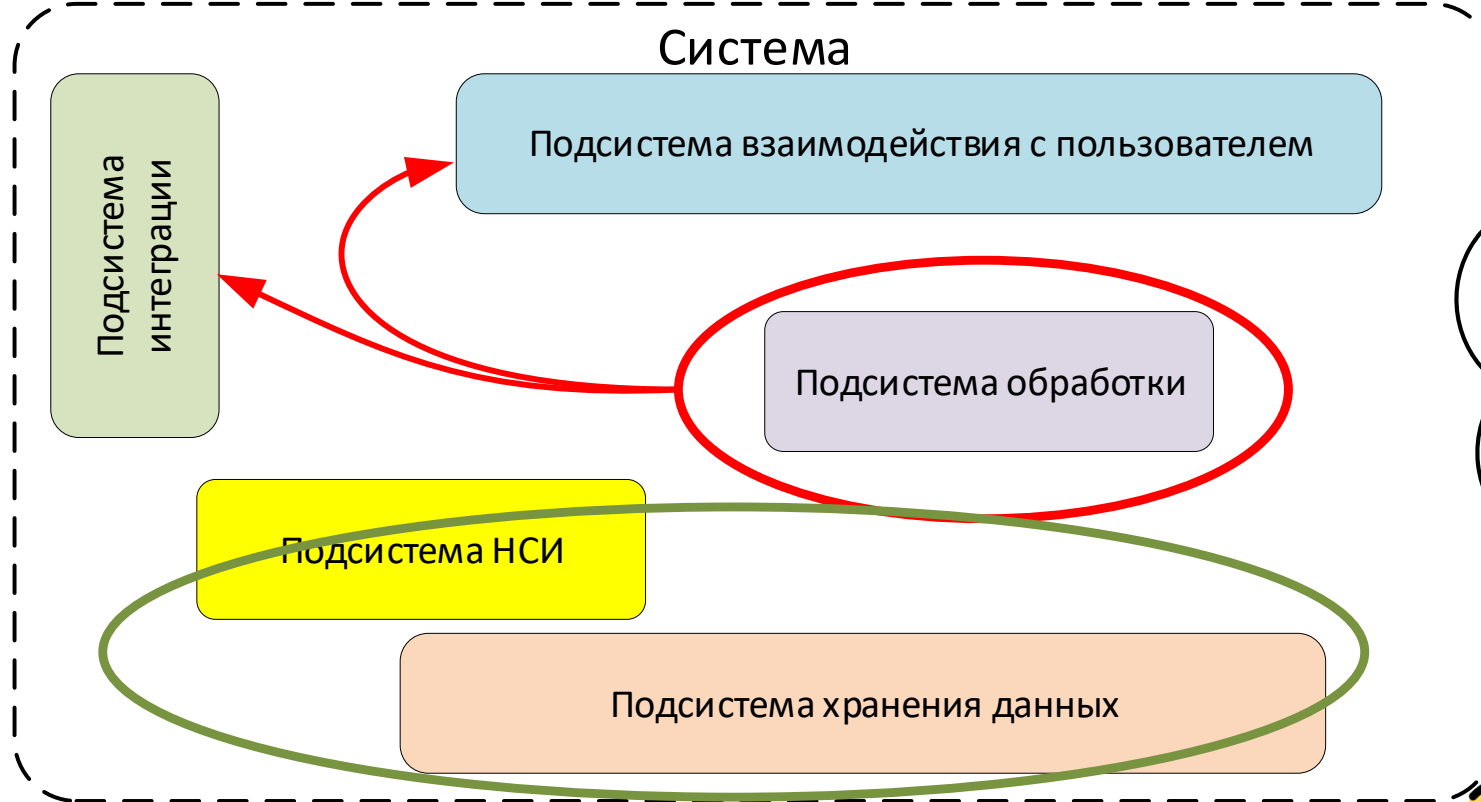
- Управляемость (независимые команды)
- Гибкость продукта (внесение изменений)
- «Усвояемость» (возможность изучения системы по частям)





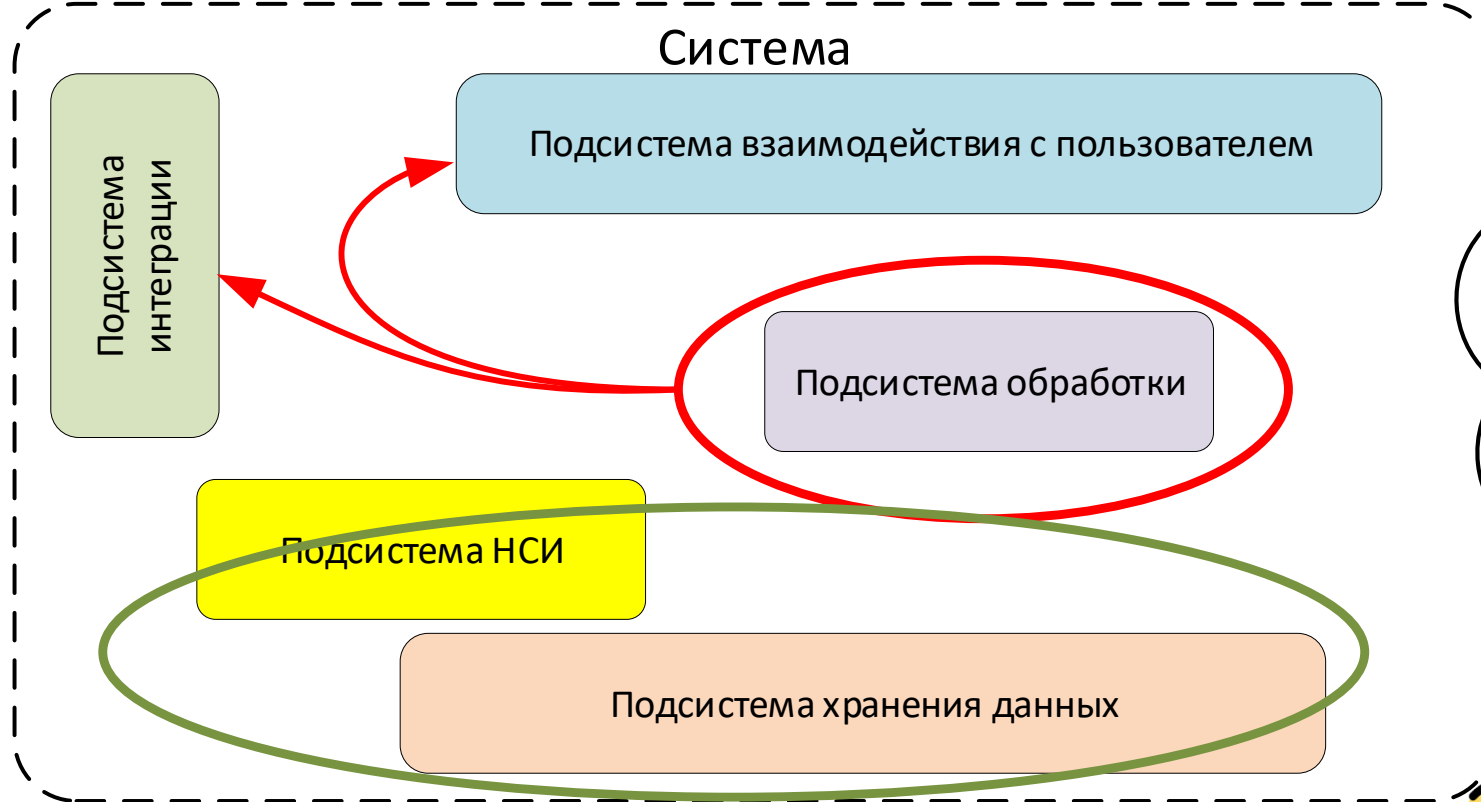
- Управляемость  
(независимые команды)
- Гибкость продукта  
(внесение изменений)
- «Усвояемость»  
(возможность изучения  
системы по частям)





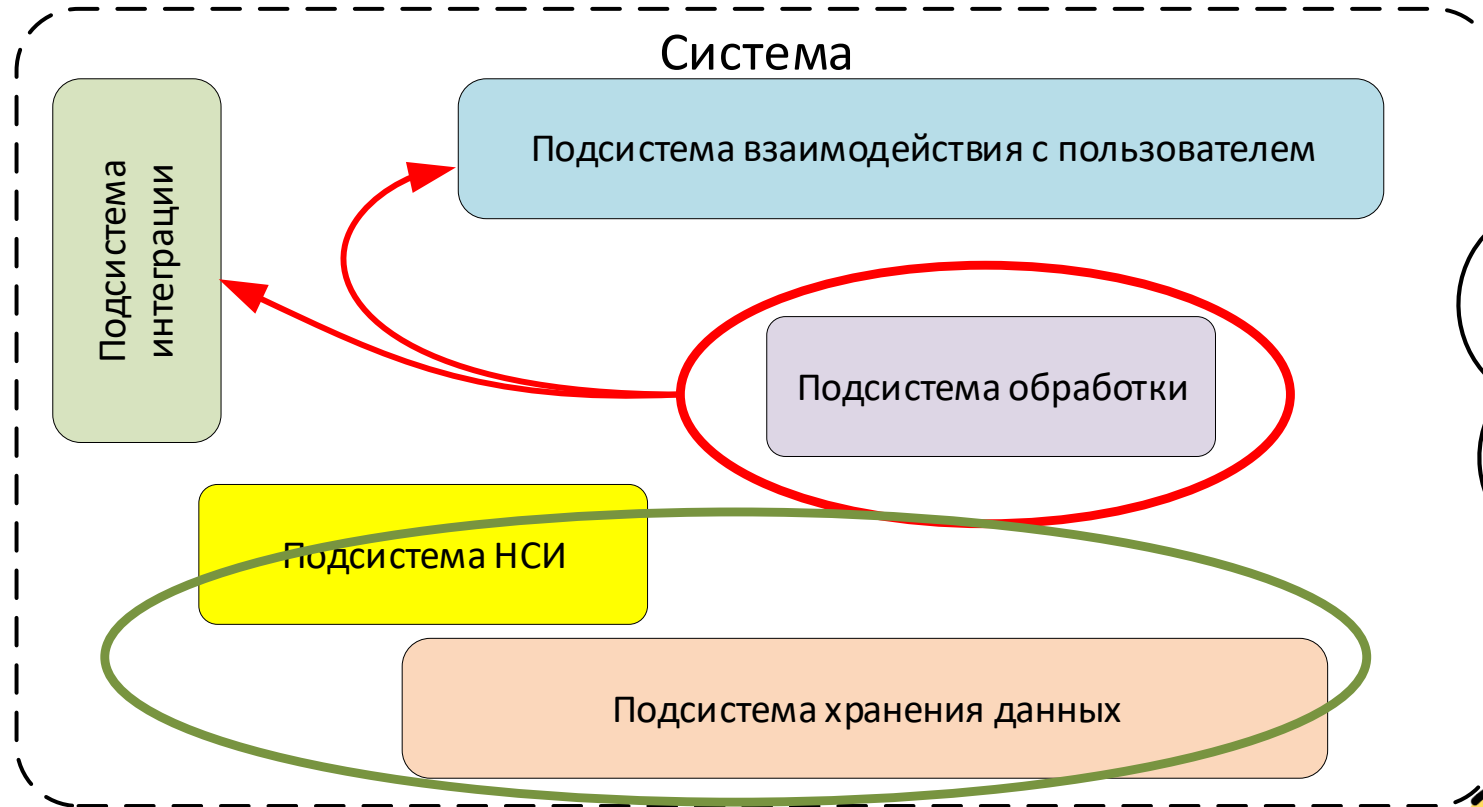
- Управляемость  
(независимые команды)
- Гибкость продукта  
(внесение изменений)
- «Усвояемость»  
(возможность изучения  
системы по частям)





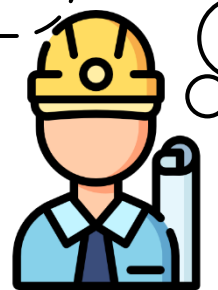
- Управляемость  
(независимые команды)
- Гибкость продукта  
(внесение изменений)
- «Усвояемость»  
(возможность изучения системы по частям)





- Управляемость  
(независимые команды)
- Гибкость продукта  
(внесение изменений)
- «Усвояемость»  
(возможность изучения системы по частям)

Декомпозиция в лучшем случае бесполезна





# ПОДХОДЫ К ДЕКОМПОЗИЦИИ

Или «как сойти за своего на тусовке архитекторов»

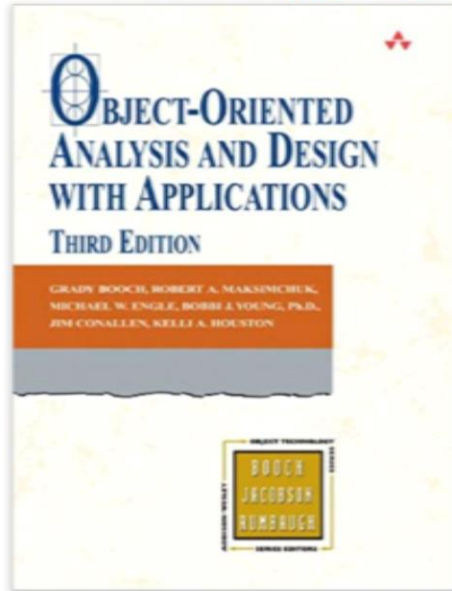


# Object-Oriented Analysis and Design with Applications (3rd Edition) 3rd Edition

by Grady Booch (Author), Robert A. Maksimchuk (Author), Michael W. Engle (Author), Bobbi J. Young (Author), Jim Conallen (Author), Kelli A. Houston (Author)

★★★★☆ 40 customer reviews

Look inside



ISBN-13: 978-0201895513

ISBN-10: 020189551X

Why is ISBN important?

Have one to sell?

Sell on Amazon

Kindle

from \$31.31

**Hardcover**

**\$37.10 - \$66.23**

Paperback

\$25.99 - \$29.26

Other Sellers

See all 6 versions

Buy used

**\$37.10**

Condition: **Used - Very Good**

35 Used from **\$37.10**

**In Stock.** Sold by Great Value Online Seller, Fulfilled by Amazon

FREE Shipping

Access codes and supplements are not guaranteed with used items.

Deliver to Maxim - Penza



Add to Cart

Buy new

\$66.23

**More Buying Choices**

56 used & new from **\$31.49**

21 New from **\$31.49** | 35 Used from **\$37.10**

See All Buying Options



College student? Get FREE shipping and exclusive deals

LEARN MORE



Не равно «язык  
программирования»!

Алгоритмическая  
декомпозиция



Объектно-  
ориентированная  
декомпозиция



## Алгоритмическая декомпозиция



## Алгоритмическая декомпозиция

- Разделение системы «вдоль» процесса функционирования (алгоритма)



## Алгоритмическая декомпозиция

- Разделение системы «вдоль» процесса функционирования (алгоритма)
- Переиспользование шагов процесса/ подпроцессов

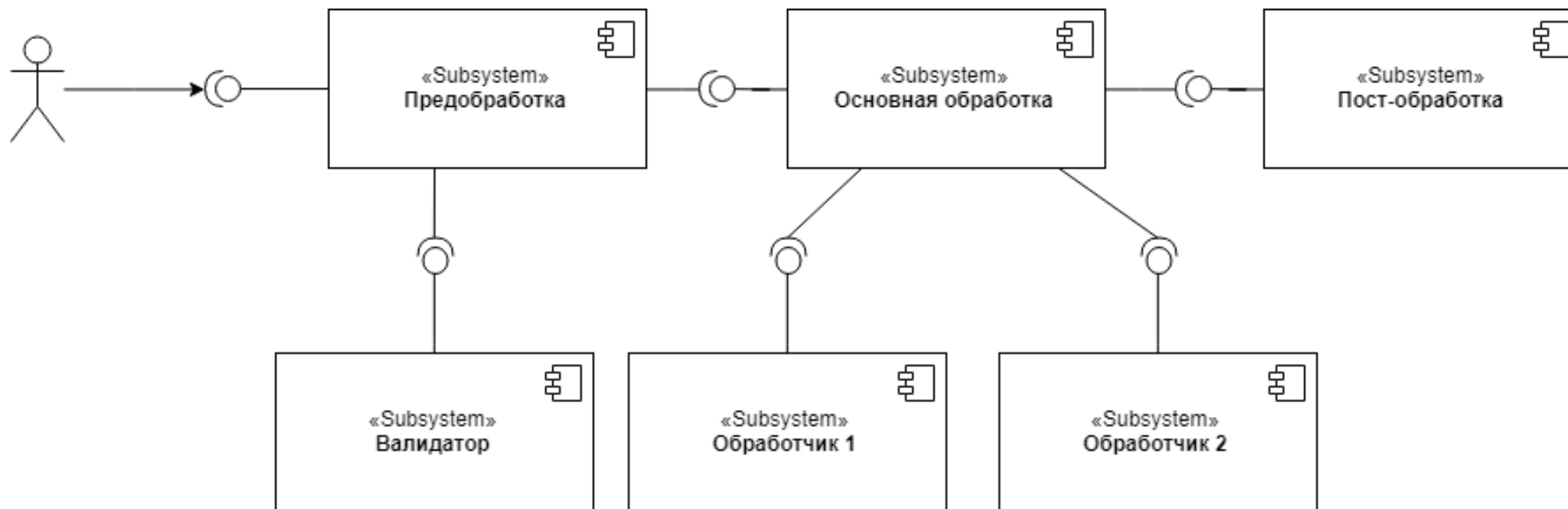


## Алгоритмическая декомпозиция

- Разделение системы «вдоль» процесса функционирования (алгоритма)
- Переиспользование шагов процесса/ подпроцессов
- Примеры:
  - Системы документооборота
  - Системы потоковой обработки данных (в том числе – IoT)
  - Системы аналитики
  - Function-as-a-Service (AWS Lambda, Azure Functions, etc.)

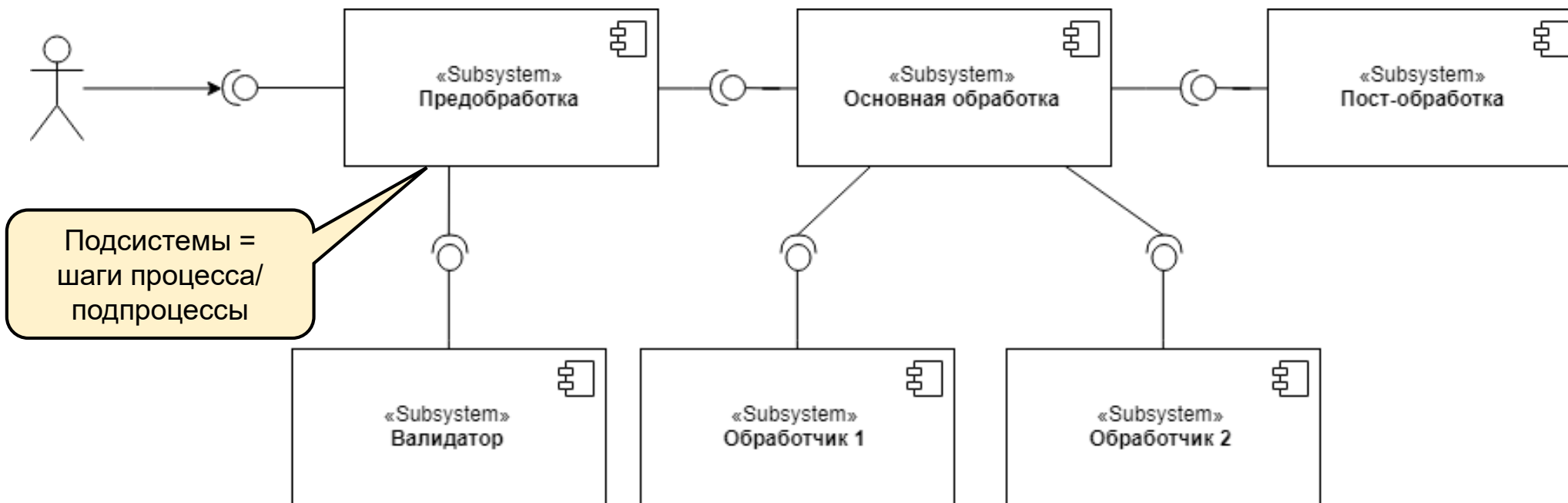


## Алгоритмическая декомпозиция



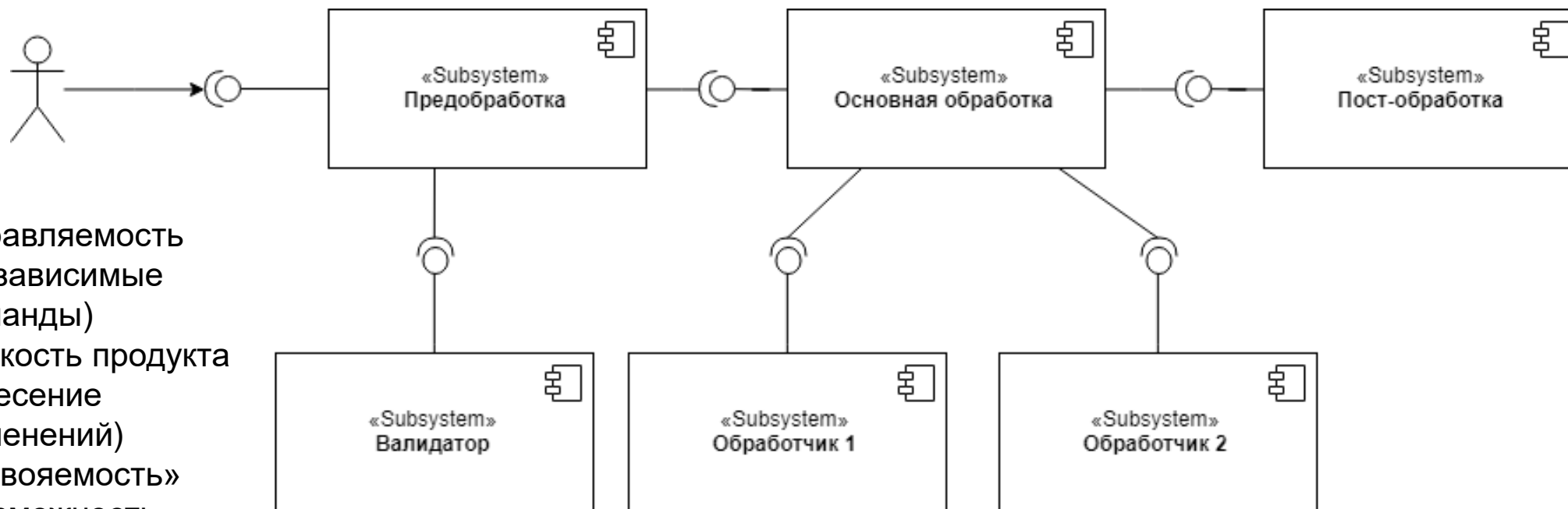


## Алгоритмическая декомпозиция





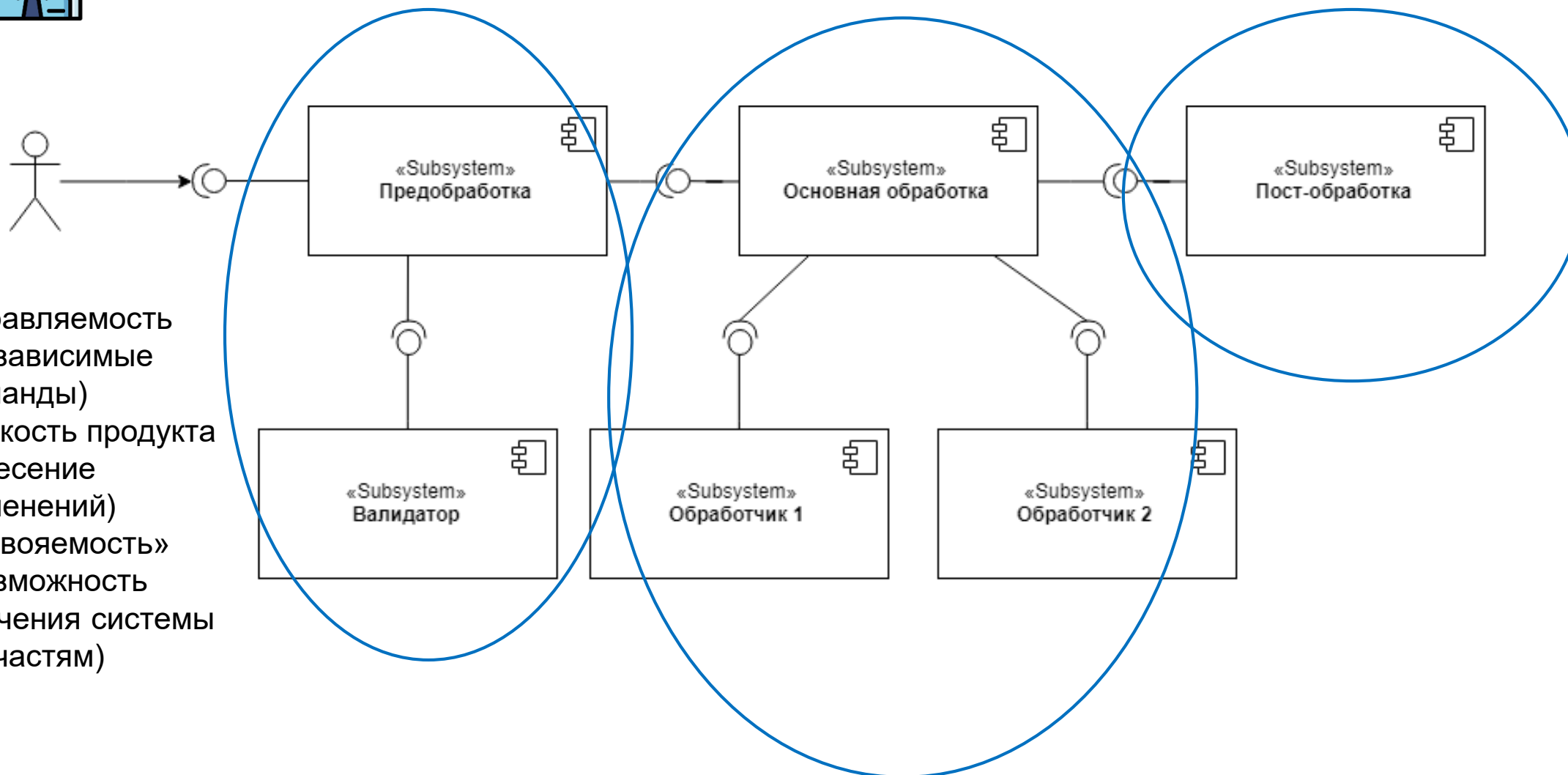
## Алгоритмическая декомпозиция



- ❑ Управляемость (независимые команды)
- ❑ Гибкость продукта (внесение изменений)
- ❑ «Усвояемость» (возможность изучения системы по частям)



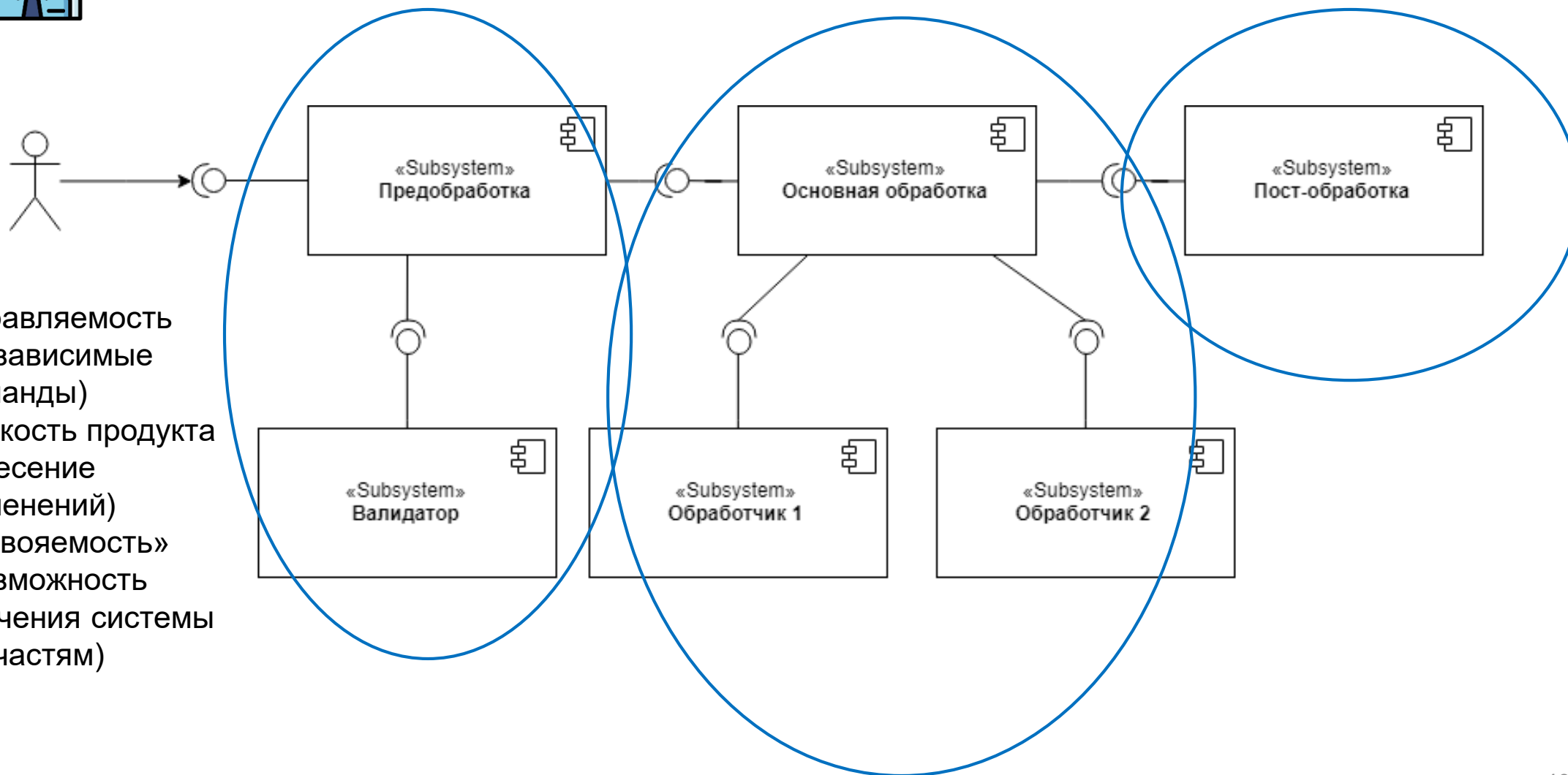
## Алгоритмическая декомпозиция



- ❑ Управляемость (независимые команды)
- ❑ Гибкость продукта (внесение изменений)
- ❑ «Усвояемость» (возможность изучения системы по частям)



## Алгоритмическая декомпозиция



Управляемость  
(независимые команды)



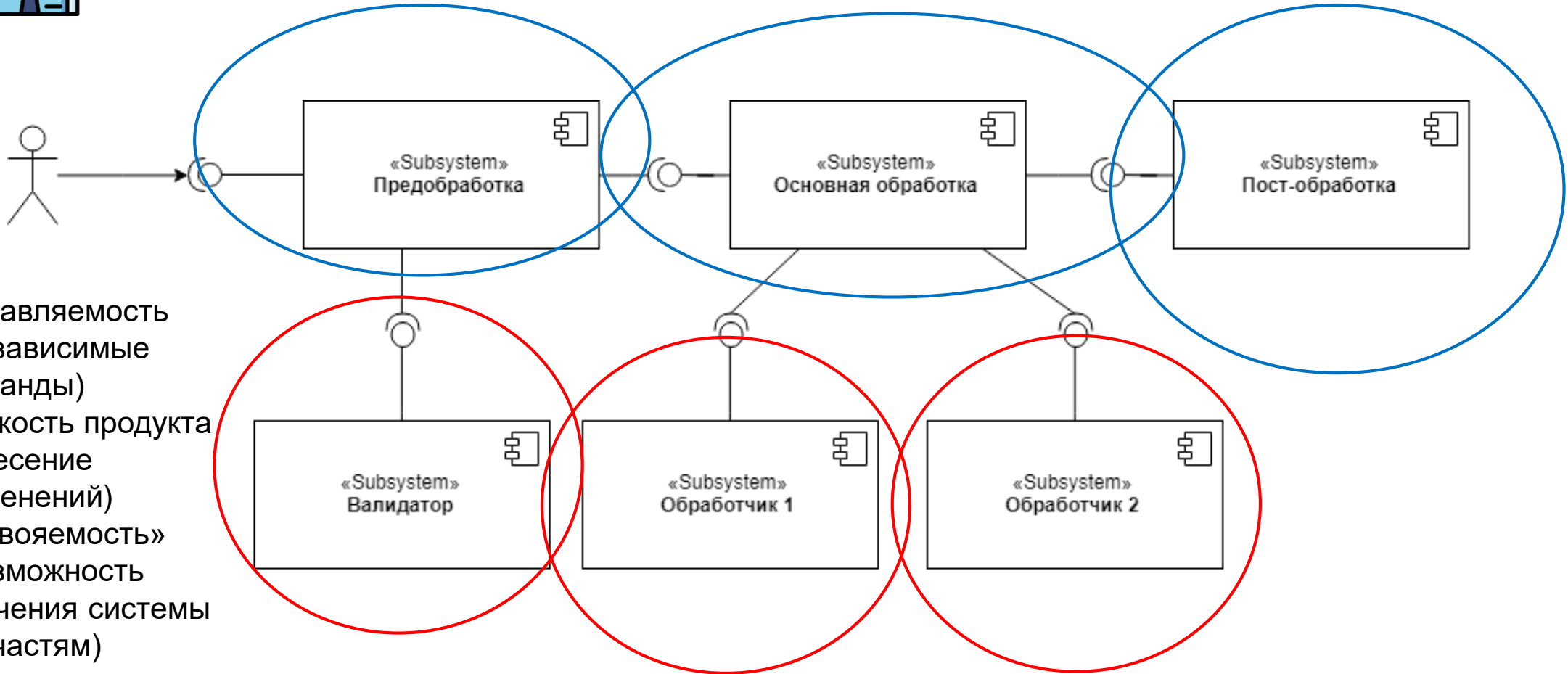
Гибкость продукта  
(внесение изменений)



«Усвояемость»  
(возможность изучения системы по частям)

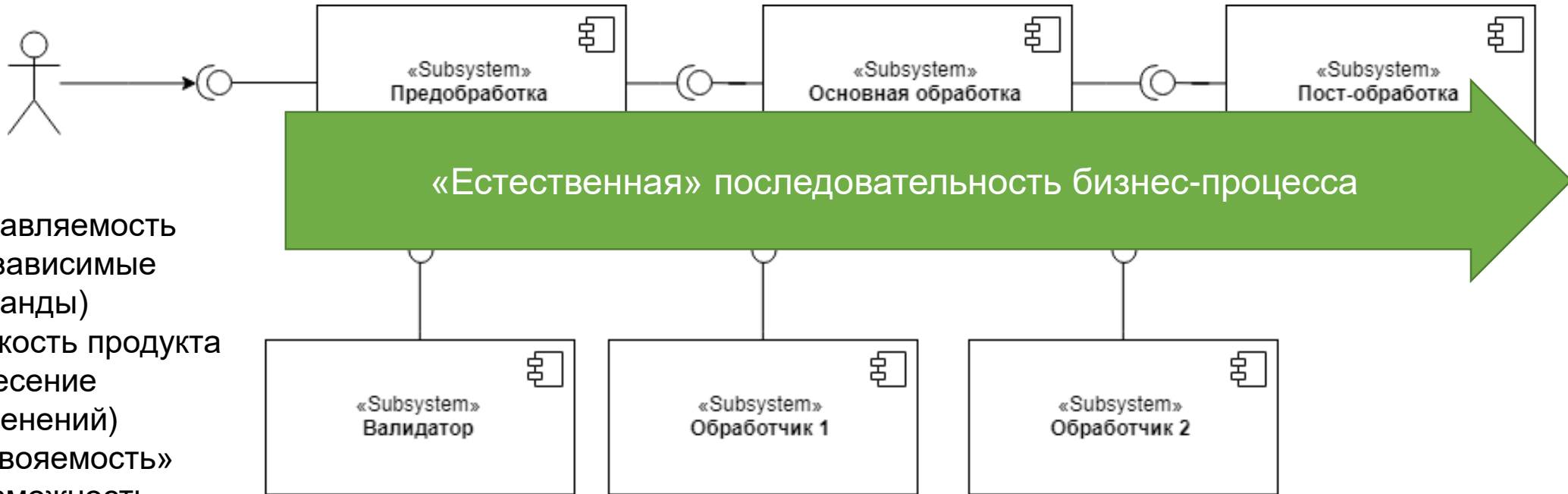


# Алгоритмическая декомпозиция





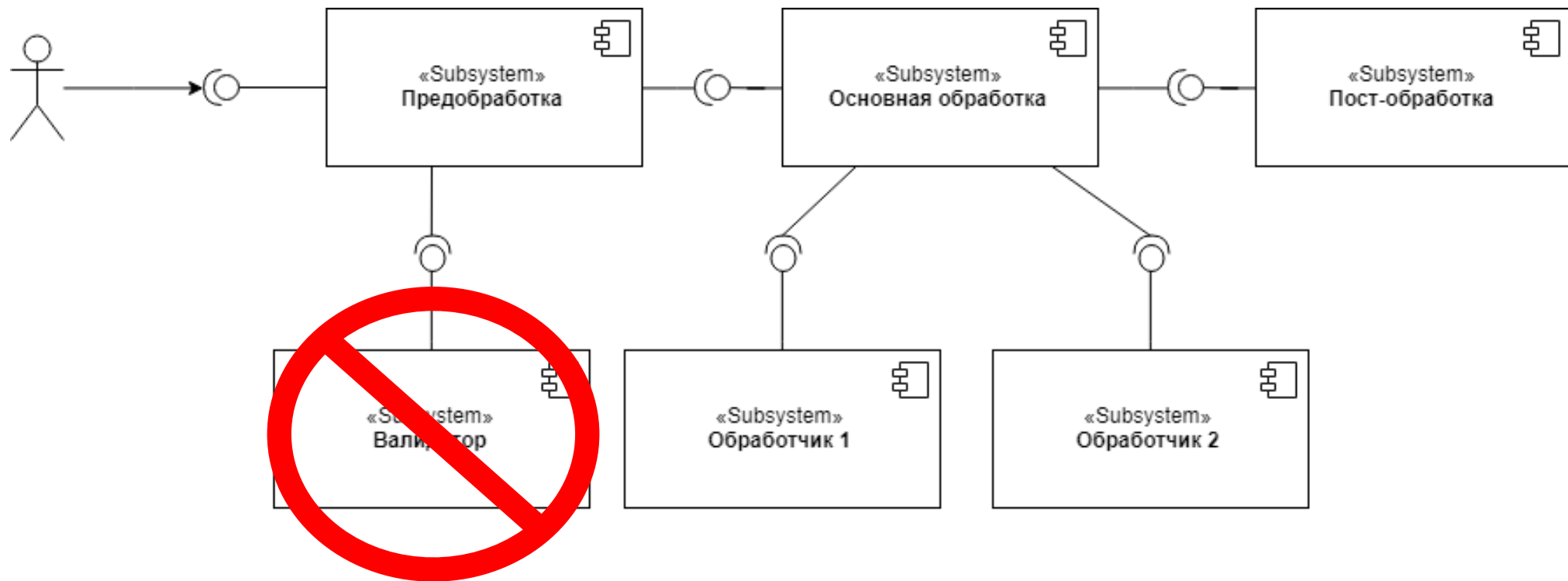
## Алгоритмическая декомпозиция



- ✓ Управляемость (независимые команды)
- ✓ Гибкость продукта (внесение изменений)
- ✓ «Усвояемость» (возможность изучения системы по частям)

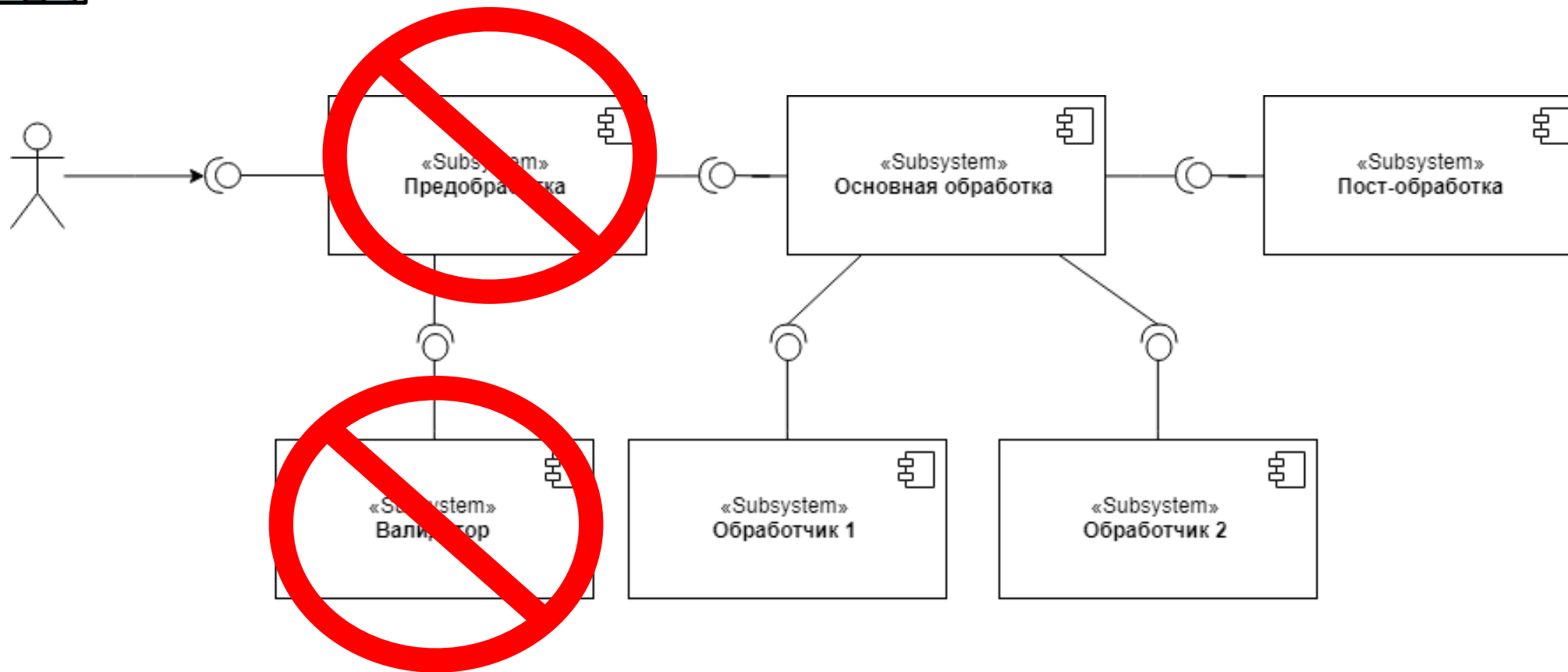


# Алгоритмическая декомпозиция



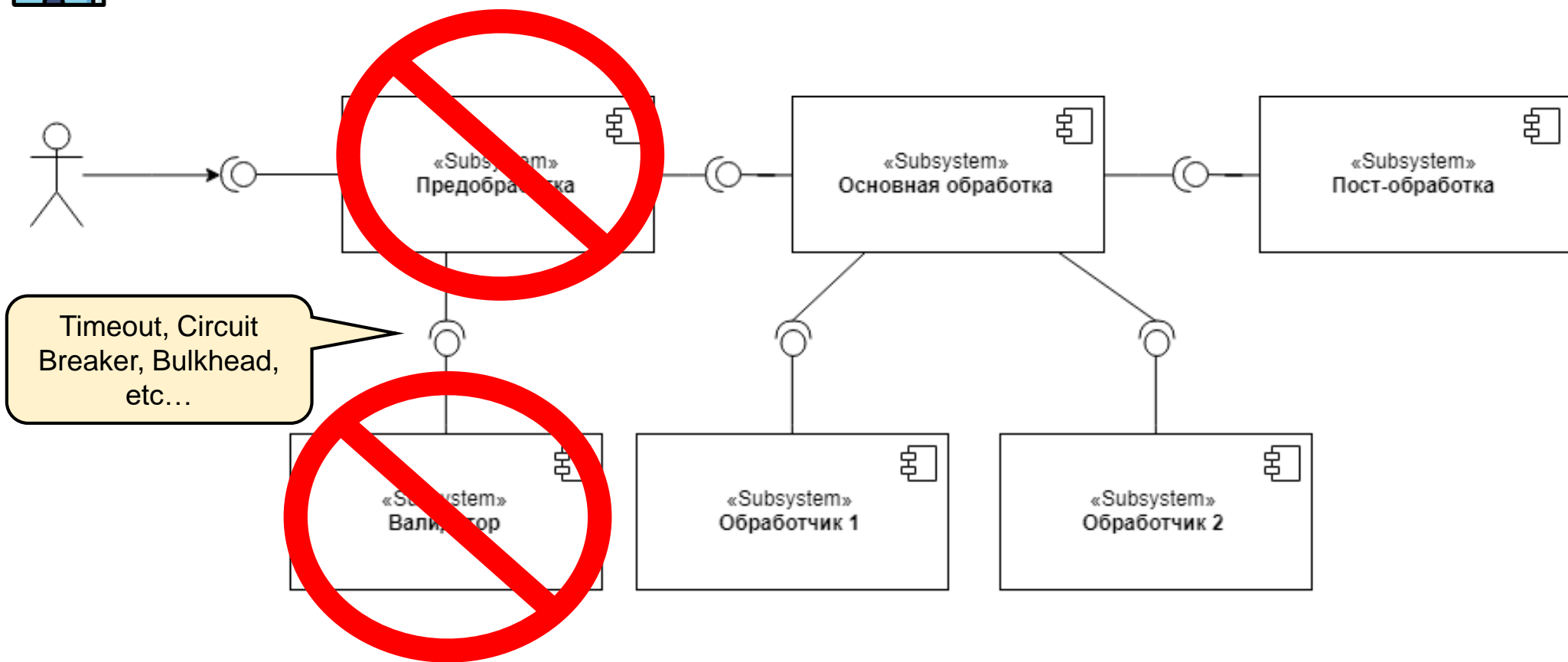


## Алгоритмическая декомпозиция





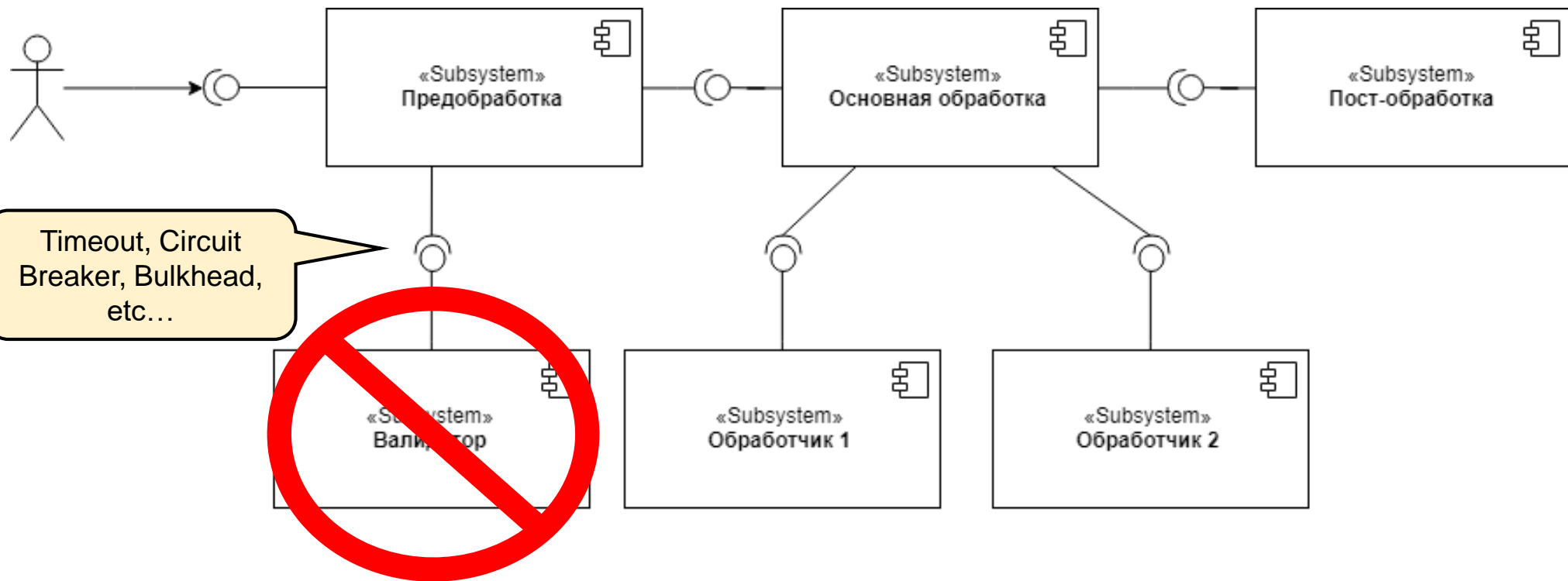
# Алгоритмическая декомпозиция





# Алгоритмическая декомпозиция

Не  
шмогла...

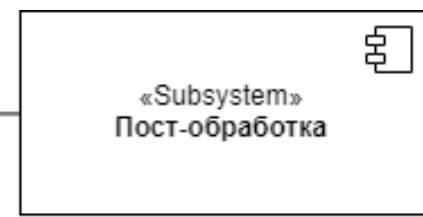
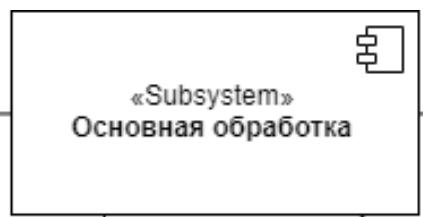
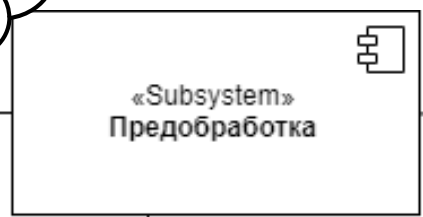
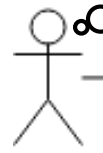




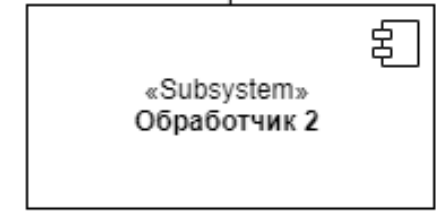
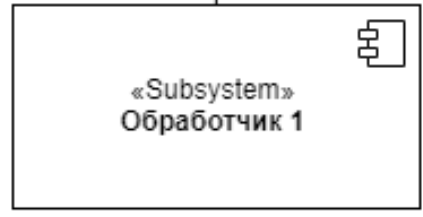
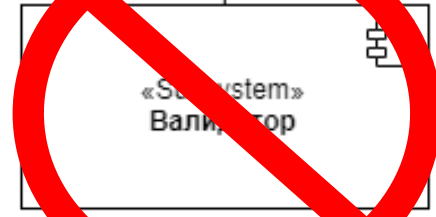
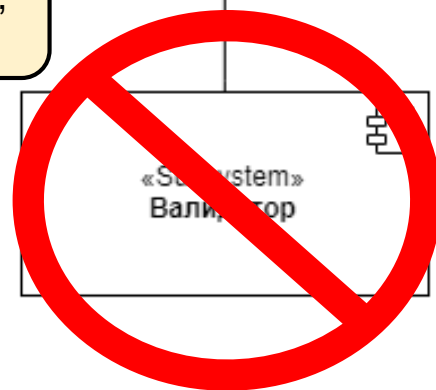
# Алгоритмическая декомпозиция



Не  
шмогла...



Timeout, Circuit  
Breaker, Bulkhead,  
etc...





## Алгоритмическая декомпозиция

- Изучать:
  - Паттерн «Pipe-and-filters»
  - Принципы работы распределенных систем
    - Асинхронный СХ!
  - Для самых технически подкованных - Streaming Architecture



## Алгоритмическая декомпозиция

- Обращать внимание:
  - Доступность → требования к отклику и времени выполнения процессов
  - Горизонтальное масштабирование → требования, которые ограничивают масштабируемость (т.е. ведут к statefull-подсистемам)



## Объектно-ориентированная декомпозиция



## Объектно-ориентированная декомпозиция

- Разделение системы на основе принадлежности ее элементов к абстракциям предметной области



## Объектно-ориентированная декомпозиция

- Разделение системы на основе принадлежности ее элементов к абстракциям предметной области
- Переиспользование абстракций и их поведения

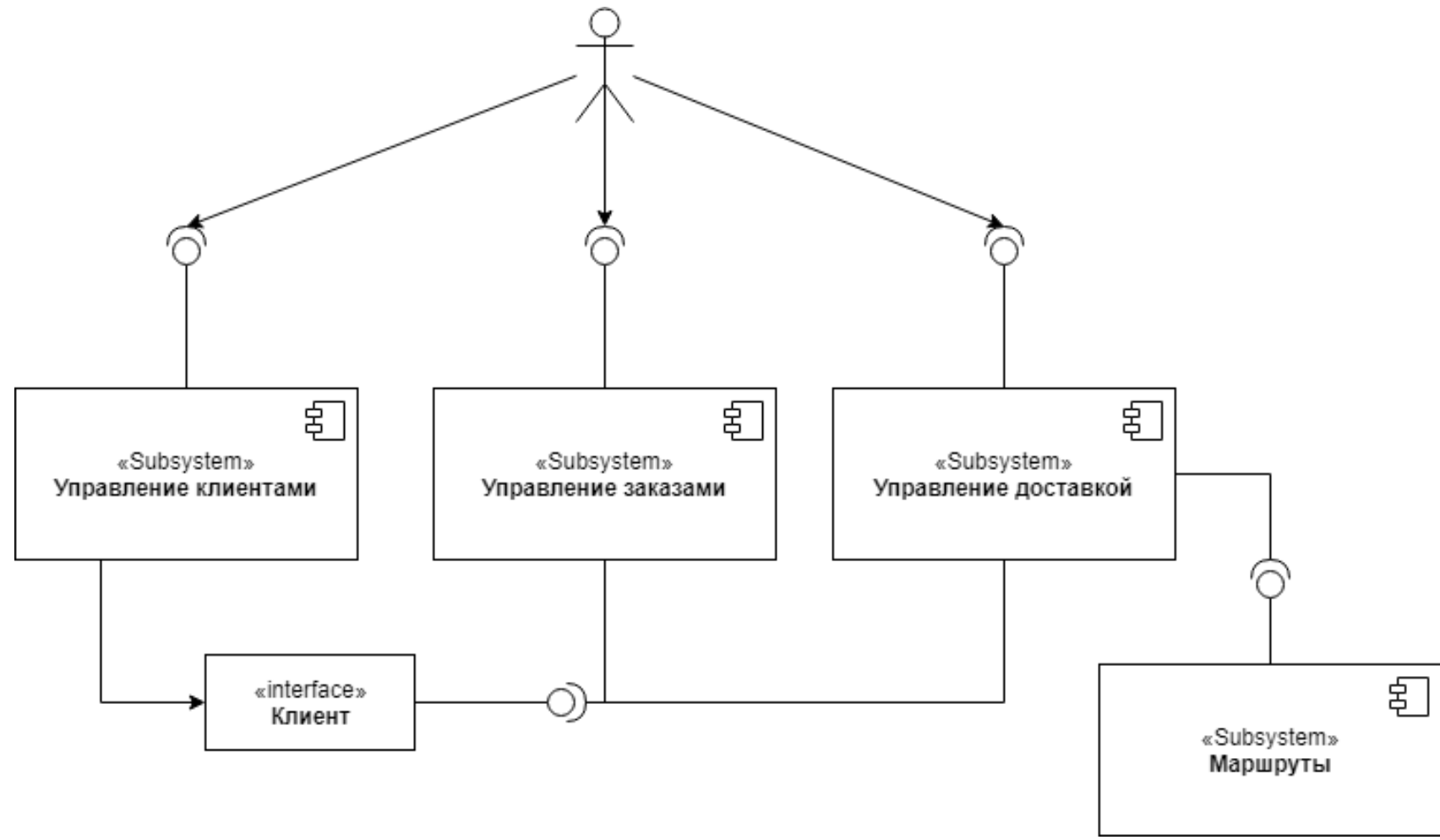


## Объектно-ориентированная декомпозиция

- Разделение системы на основе принадлежности ее элементов к абстракциям предметной области
- Переиспользование абстракций и их поведения
- Примеры:
  - Системы-реестры
  - Системы бэк-офиса предприятия
  - «Классические» микросервисы

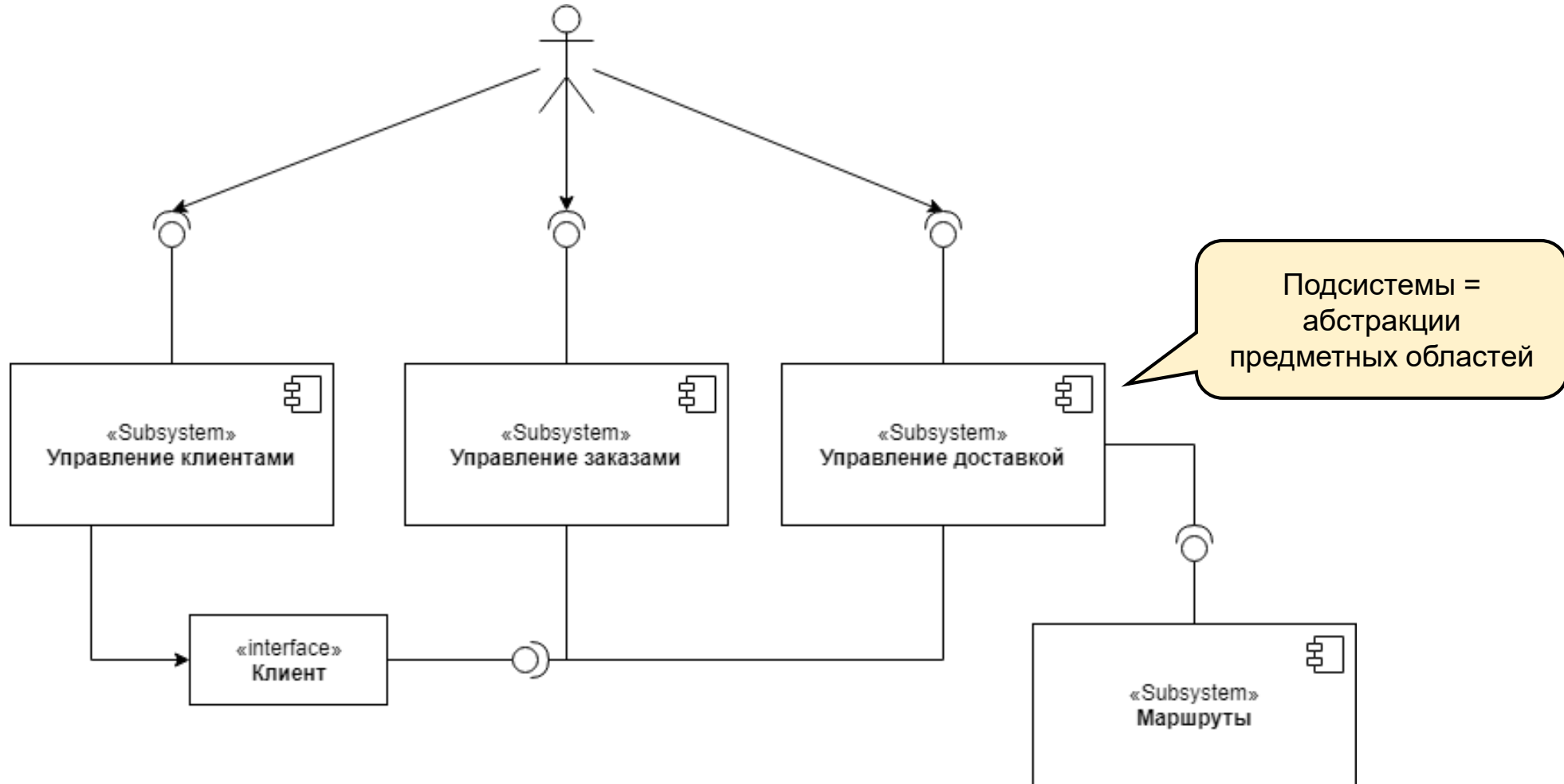


# Объектно-ориентированная декомпозиция



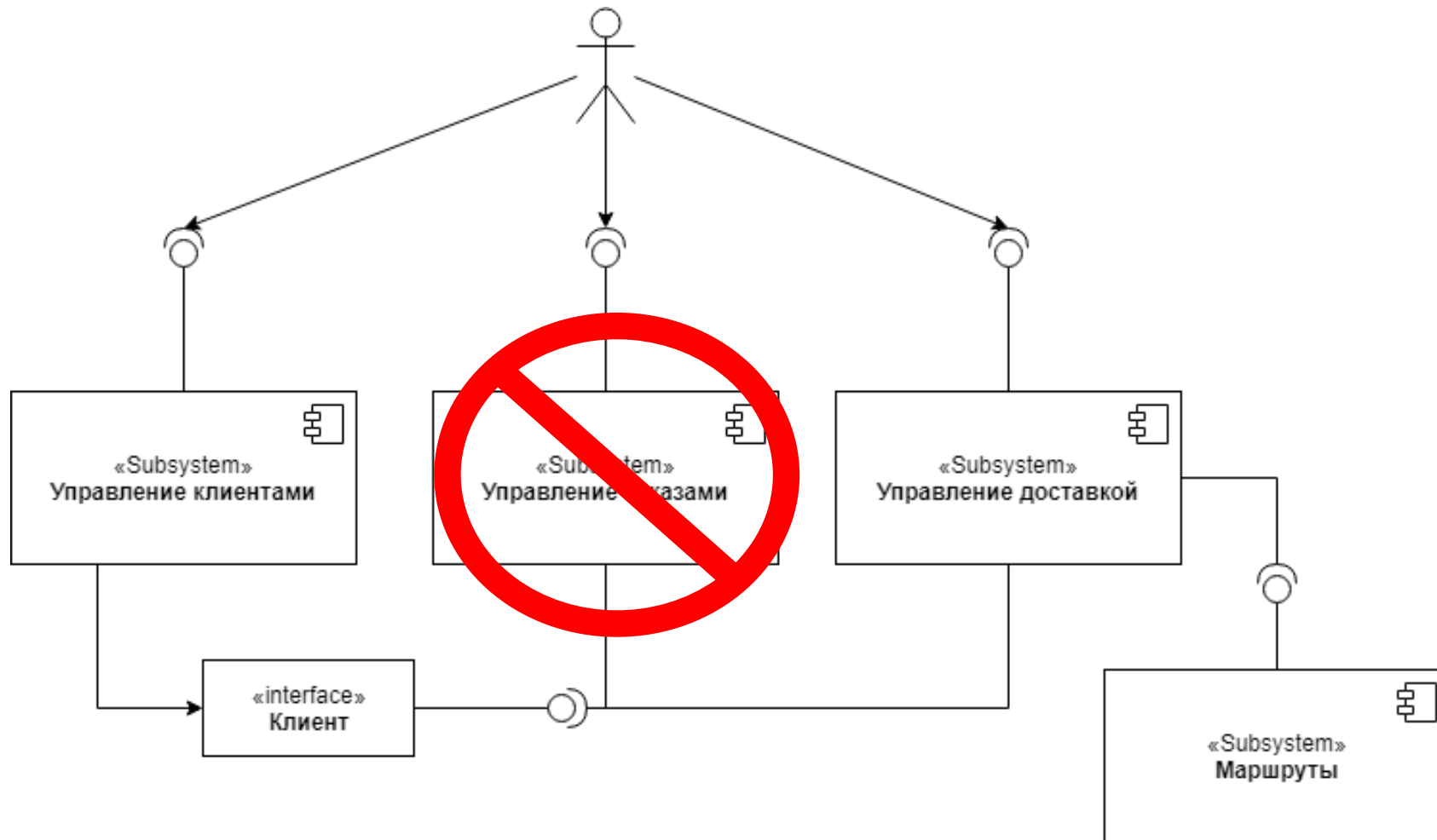


# Объектно-ориентированная декомпозиция



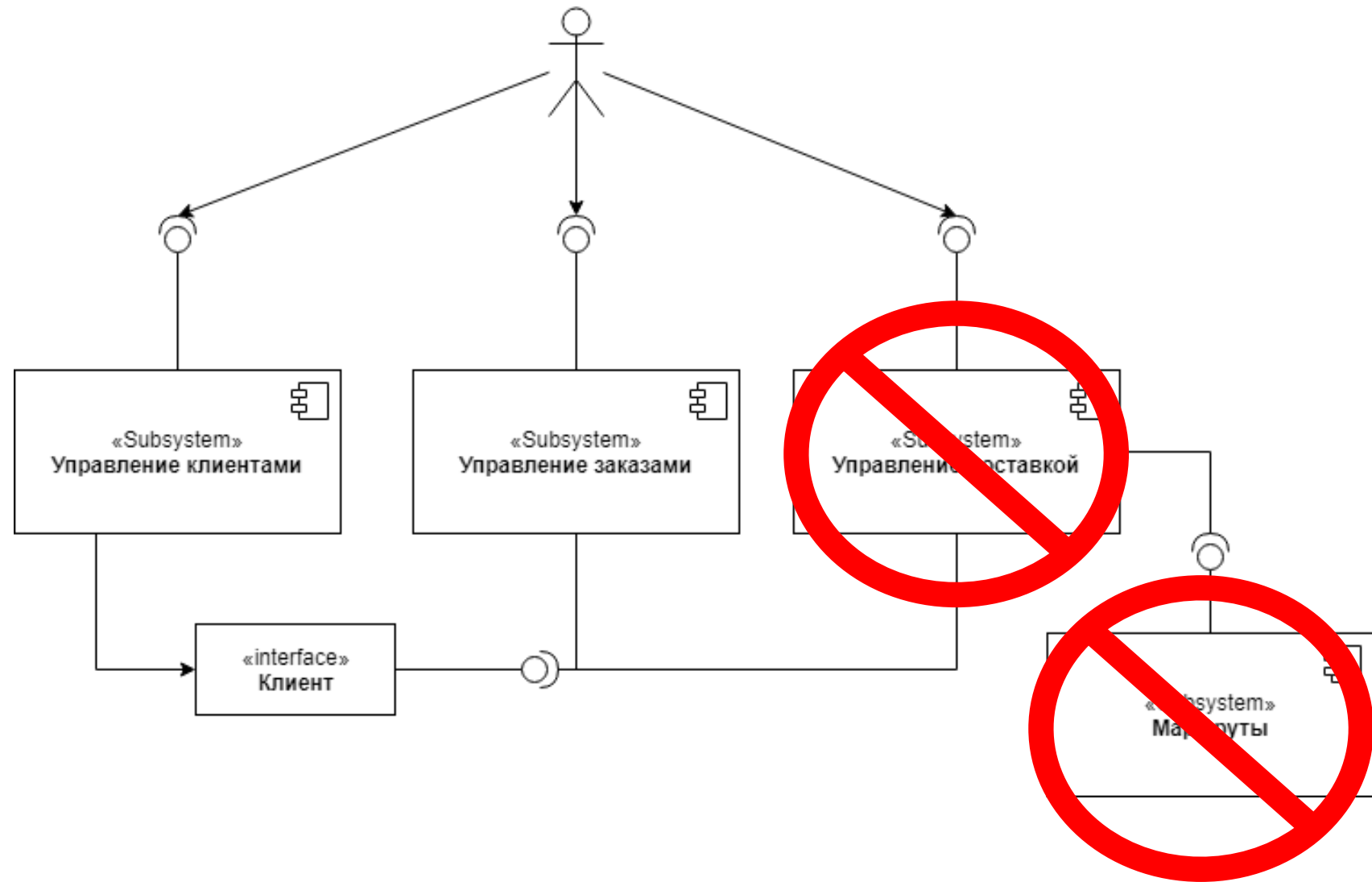


## Объектно-ориентированная декомпозиция



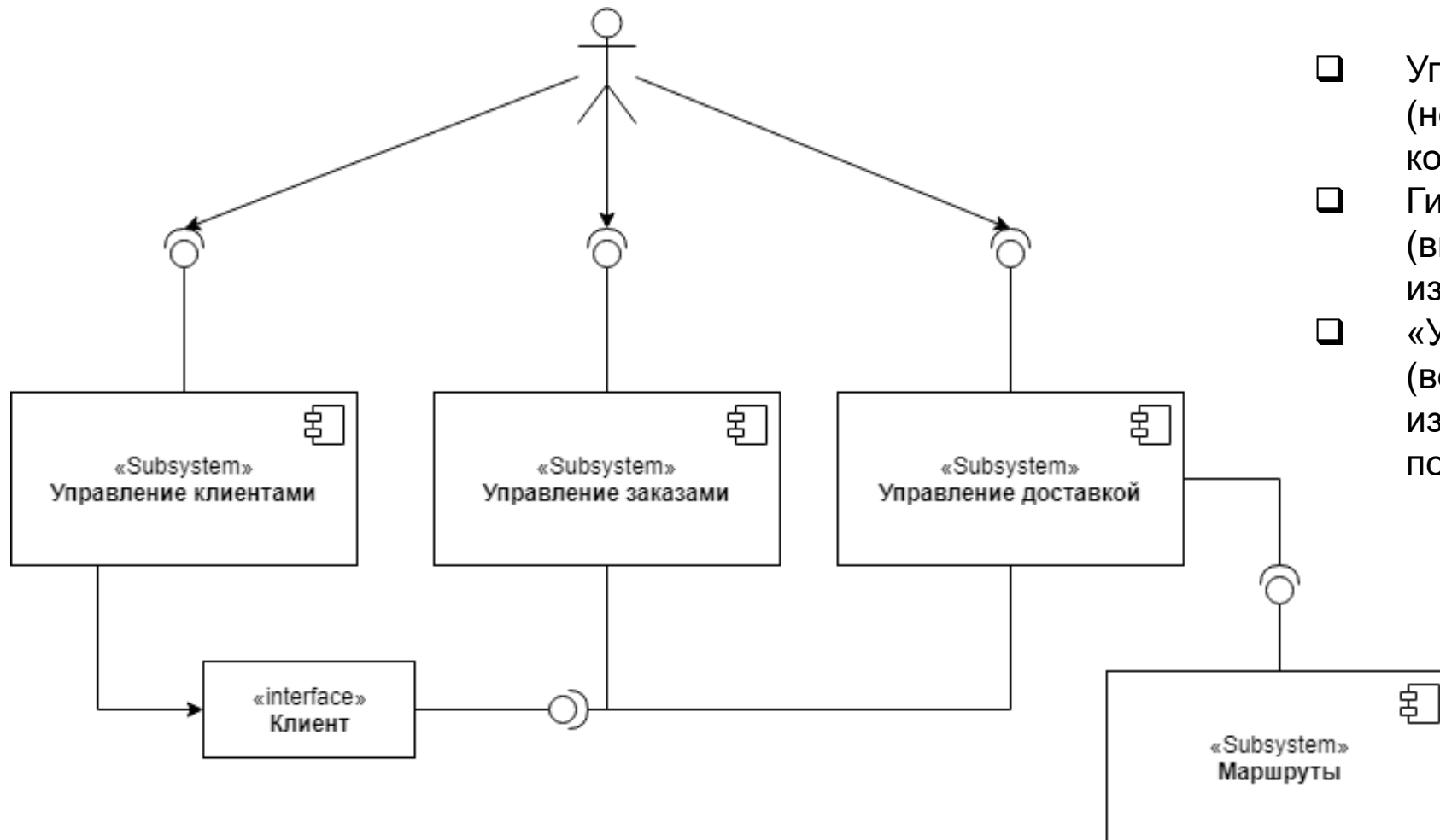


# Объектно-ориентированная декомпозиция





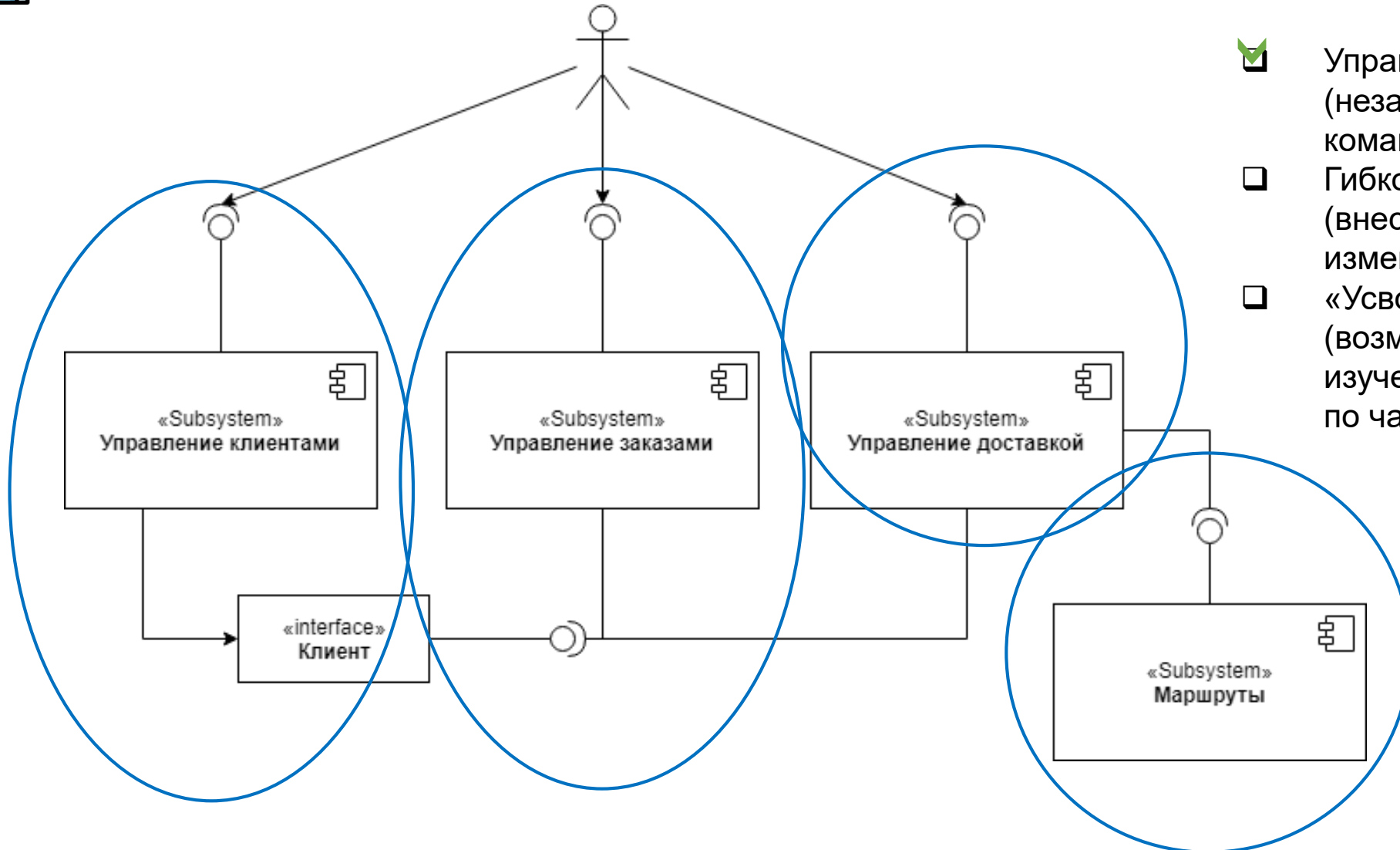
## Объектно-ориентированная декомпозиция



- ❑ Управляемость (независимые команды)
- ❑ Гибкость продукта (внесение изменений)
- ❑ «Усвояемость» (возможность изучения системы по частям)



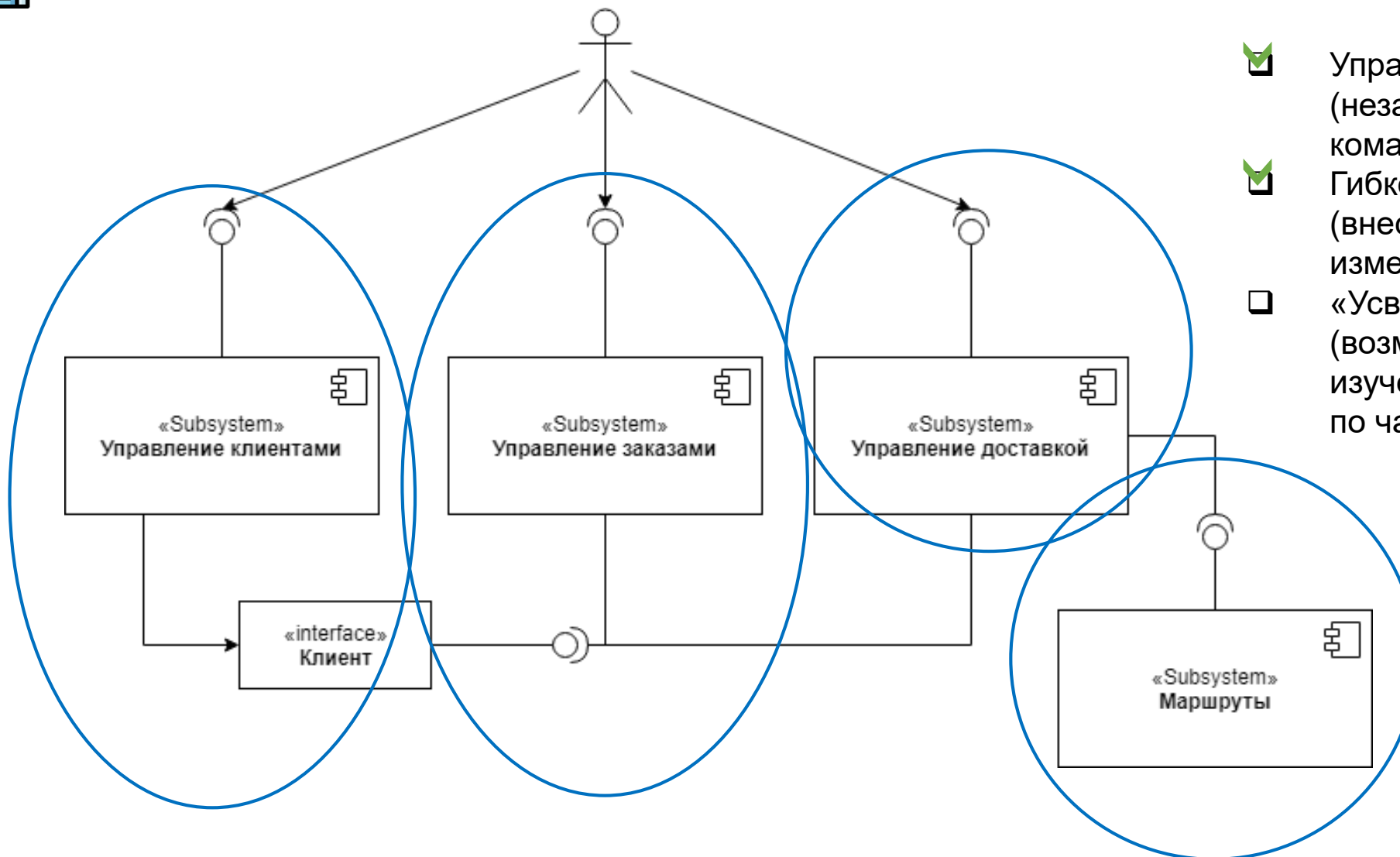
## Объектно-ориентированная декомпозиция



- ✓ Управляемость (независимые команды)
- Гибкость продукта (внесение изменений)
- «Усвояемость» (возможность изучения системы по частям)



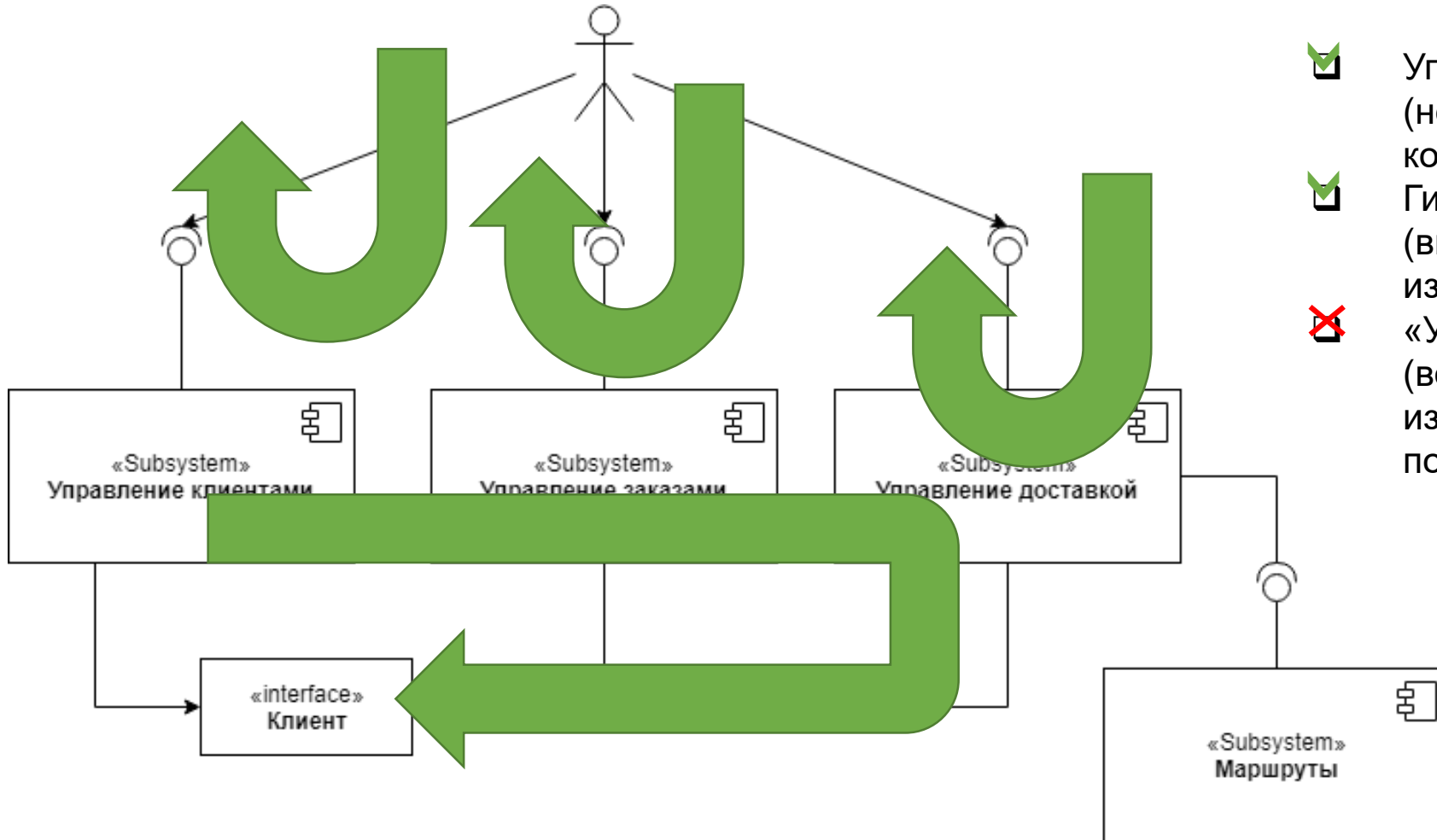
## Объектно-ориентированная декомпозиция



- ✓ Управляемость (независимые команды)
- ✓ Гибкость продукта (внесение изменений)
- ❑ «Усвояемость» (возможность изучения системы по частям)



## Объектно-ориентированная декомпозиция



- ✓ Управляемость (независимые команды)
- ✓ Гибкость продукта (внесение изменений)
- ✗ «Усвояемость» (возможность изучения системы по частям)



## Объектно-ориентированная декомпозиция

- Изучать:
  - Domain-Driven Design
  - Принципы работы распределенных систем



## Объектно-ориентированная декомпозиция

- **Обращать внимание:**
  - Организационные границы (зоны ответственности бизнеса, зоны ответственности разработки и т.д.)
  - Концептуальная целостность



Алгоритмическая  
декомпозиция

Объектно-  
ориентированная  
декомпозиция





Алгоритмическая  
декомпозиция

Объектно-  
ориентированная  
декомпозиция

Нужно выбрать  
основную  
парадигму





Алгоритмическая  
декомпозиция

Объектно-  
ориентированная  
декомпозиция

Нужно выбрать  
основную  
парадигму



Можно  
комбинировать, но  
одна должна быть  
основной!



# А дальше что?

Как выделять подсистемы?



# Принципы декомпозиции





# Принципы декомпозиции

- Принципы высокой сцепленности (связность) и слабой связанности («high cohesion, low coupling»)





# Принципы декомпозиции

- Принципы высокой сцепленности (связность) и слабой связанности («high cohesion, low coupling»)
- Учет границ («separation of concerns»):
  - Бизнеса
  - Организации-пользователя
  - Организации-разработчика
  - Технических процессов





# Принципы декомпозиции

- Принципы высокой сцепленности (связность) и слабой связанности («high cohesion, low coupling»)
- Учет границ («separation of concerns»):
  - Бизнеса
  - Организации-пользователя
  - Организации-разработчика
  - Технических процессов
- Сохранение концептуальной целостности (привет, Брукс!):
  - Решение решает проблему
  - Концепции системы согласованы друг с другом
  - Все части системы «реализуют одну идею»





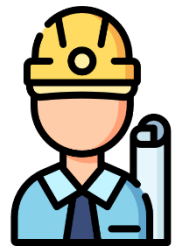
# Антипаттерны декомпозиции





# Антипаттерны декомпозиции

- Подсистема-инфраструктурный элемент (пример: «Подсистема хранения данных»)





# Антипаттерны декомпозиции

- Подсистема-инфраструктурный элемент (пример: «Подсистема хранения данных»)
- Подсистема-бог или супер-подсистема (пример: «Подсистема обработки», «Подсистема бизнес-логики»)





# Антипаттерны декомпозиции

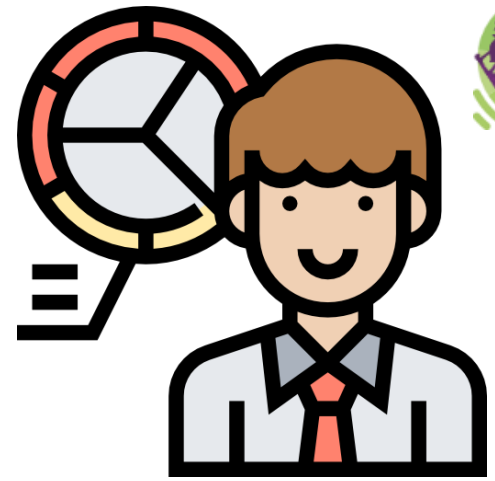
- Подсистема-инфраструктурный элемент (пример: «Подсистема хранения данных»)
- Подсистема-бог или супер-подсистема (пример: «Подсистема обработки», «Подсистема бизнес-логики»)
- Ничего кроме data-flow-подсистем





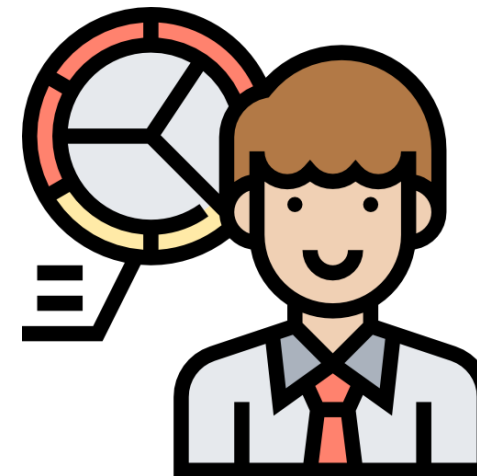
# Заключение

Или что бы я хотел, чтобы вы унесли с собой





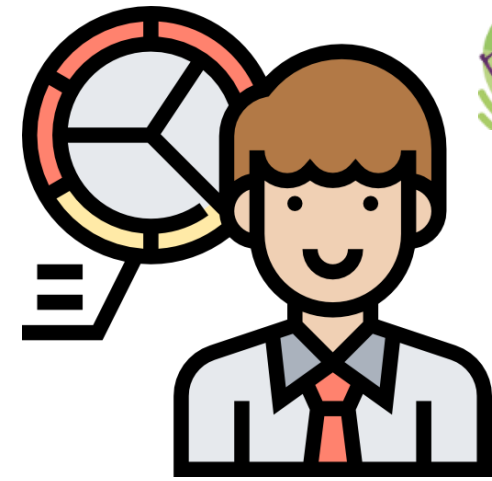
Основные критерии хорошей  
декомпозиции (чек-лист из трех  
позиций)





Основные критерии хорошей декомпозиции (чек-лист из трех позиций)

Глобальные подходы к декомпозиции системы (два варианта, которые можно комбинировать)

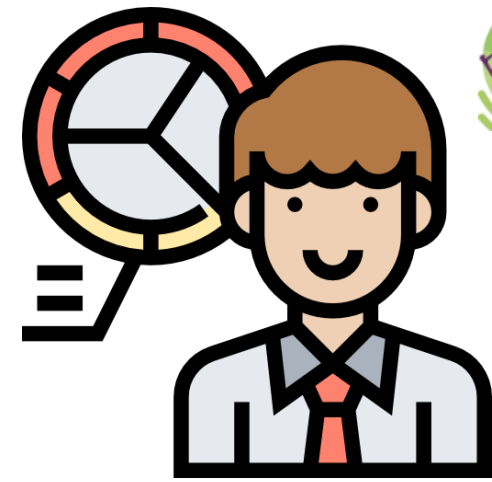




Основные критерии хорошей декомпозиции (чек-лист из трех позиций)

Глобальные подходы к декомпозиции системы (два варианта, которые можно комбинировать)

Общие принципы и антипаттерны выделения подсистем, независимо от выбранного подхода (меньше десятка)



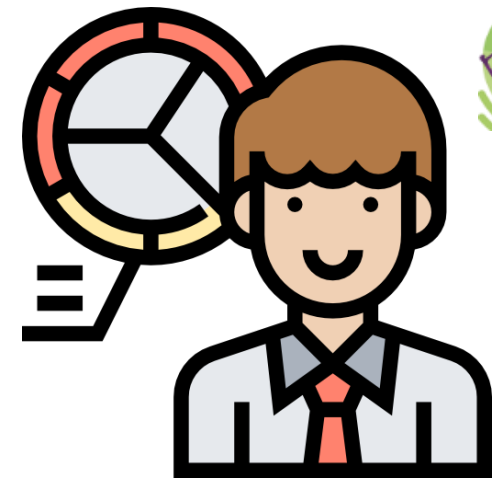


Основные критерии хорошей декомпозиции (чек-лист из трех позиций)

Глобальные подходы к декомпозиции системы (два варианта, которые можно комбинировать)

Общие принципы и антипаттерны выделения подсистем, независимо от выбранного подхода (меньше десятка)

Спасибо, добрый человек!

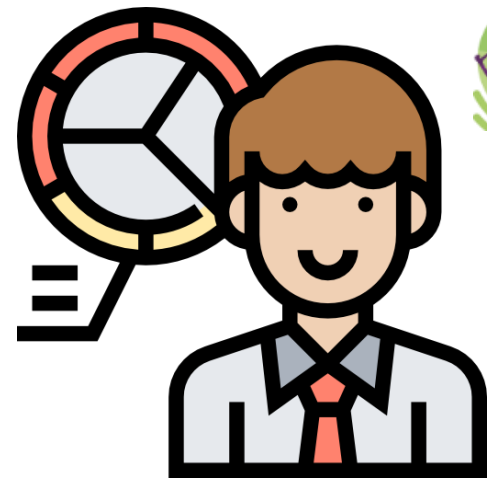




Основные критерии хорошей декомпозиции (чек-лист из трех позиций)

Глобальные подходы к декомпозиции системы (два варианта, которые можно комбинировать)

Общие принципы и антипаттерны выделения подсистем, независимо от выбранного подхода (меньше десятка)



Спасибо, добрый человек!



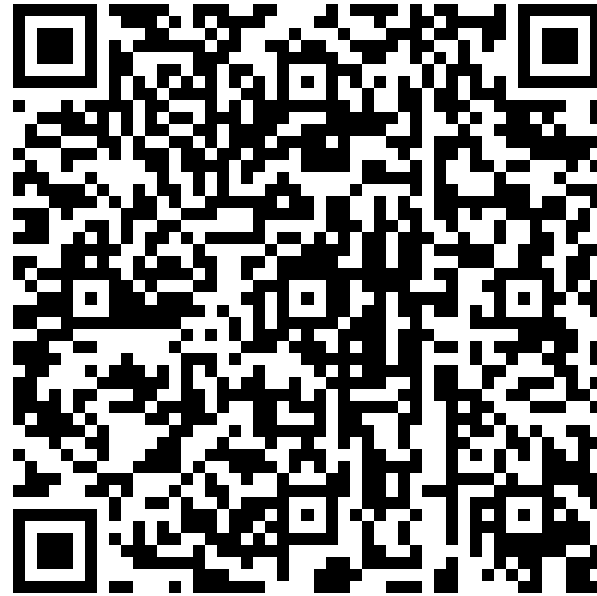
Отлично, теперь архитектор нам не нужен! Уходи, бородатый!



# ~~Задание на лето~~ Полезное ЧТИВО:

- «Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives» N. Rozanski, E. Woods
  - Наиболее полезна глава Functional Viewpoint про «настоящую» функциональную декомпозицию
  - Также читать главу архитектурные стили, там есть Pipe-and-Filters
- “On the Criteria To Be Used in Decomposing Systems into Modules”, D.L. Parnas, CMU, 1972
  - Ключевая мысль ведет к гибридной декомпозиции
- Domain-Driven Design (Эванс – концепция, Вернон – реализация)
  - must have как минимум в части контекстов, интерфейсных сущностей и агрегатов
- «Release It!: Design and Deploy Production-Ready Software (Pragmatic Programmers)», M.T. Nygard
  - Технические детали можно пропустить, читать только про паттерны





# Спасибо за внимание

Сильно не бейте



# Управление

- Зависимостей между подсистемами нет, либо они понятны
- На основе этого можно:
  - Выстроить план создания системы
  - Выстроить бюджет создания системы
  - Распределить разработку между командами, в т.ч. Масштабироваться



# Гибкость продукта (системы)

- Зависимости – аналогично предыдущему пункту
- За счет этого можно:
  - Независимо дорабатывать/ разрабатывать отдельные подсистемы
  - Разрабатывать подсистемы с разной степенью проработанности требований/ архитектурных решений
  - Эксплуатировать (мониторинг, сбор информации, развитие, DevOps)



# «УСВОЯЕМОСТЬ» СИСТЕМЫ

- Система описана (и построена!) в виде отдельных подсистем, взаимодействующих между собой через четко определенные интерфейсы, выполняющих четко определенные функции и т.д.
- Ускоряет ввод новых членов команды
- Ускоряет ввод продукта в эксплуатацию