

## Domain-driven design

## Как жить в проекте без требований



Максим Цепков

Главный архитектор  
решений CUSTIS  
Навигатор по миру Agile и  
бирюзовых организаций

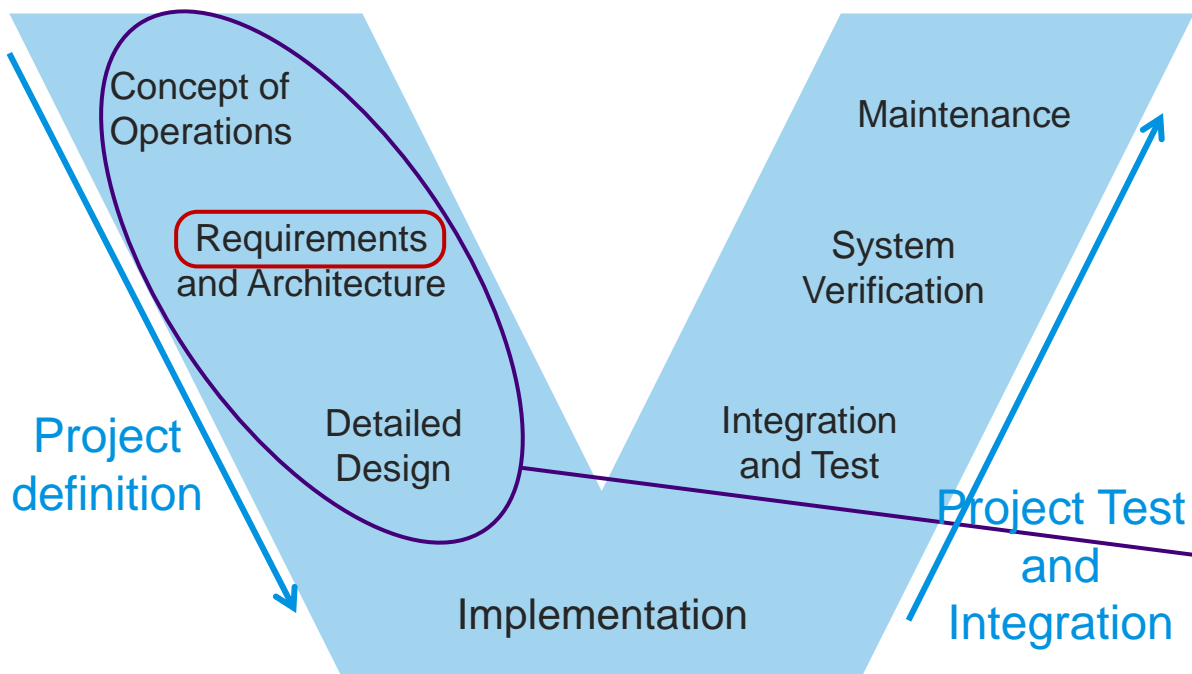
<http://mtsepkov.org>

Юрий Куприянов

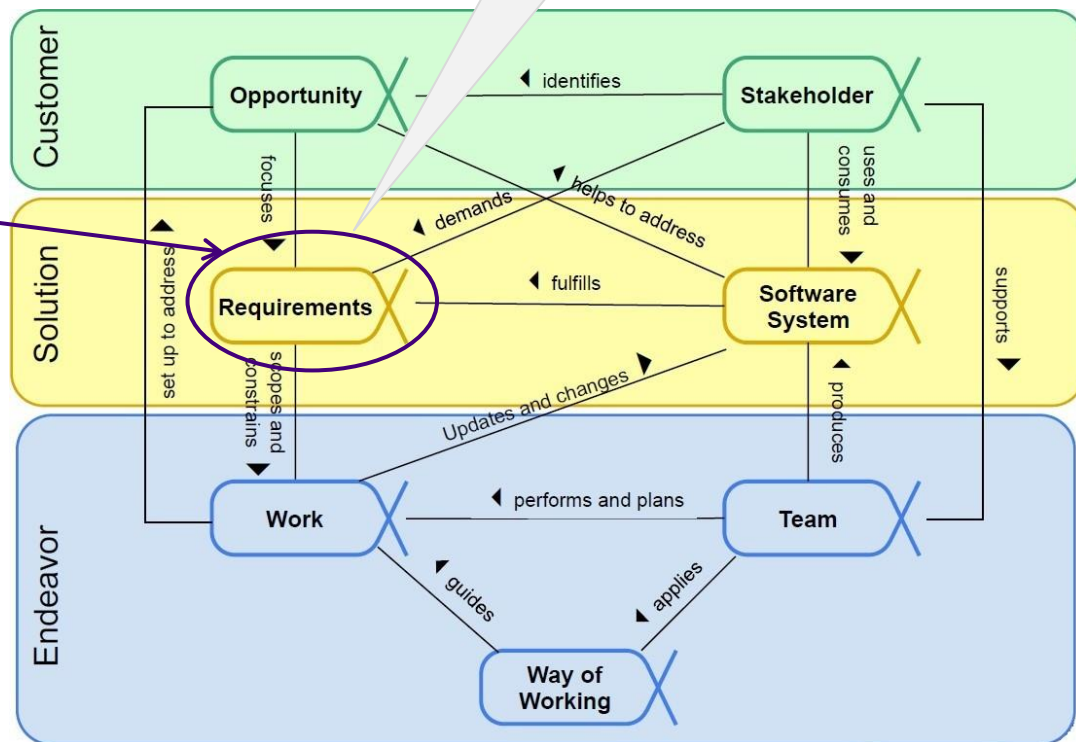
Директор по продуктам  
Школы системного анализа  
и проектирования  
Systems.Education



# Что такое «требования»: два смысла



В [OMG Essence](#) все это называли «требования», изменив смысл термина

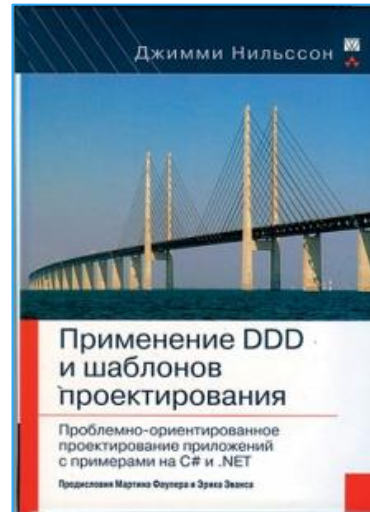


# DDD – источники



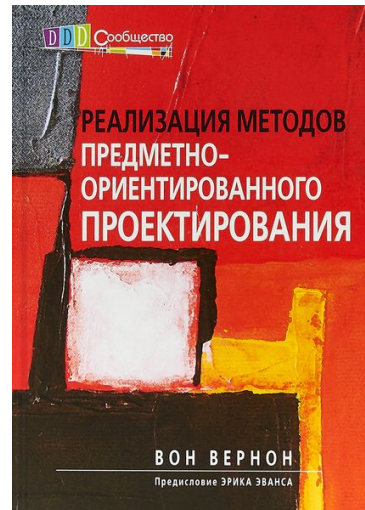
## Концептуальная книга Эрика Эванса

- на английском – в 2003 г.
- на русском – только в 2010 г.



## Практическая книга Джимми Нильссона

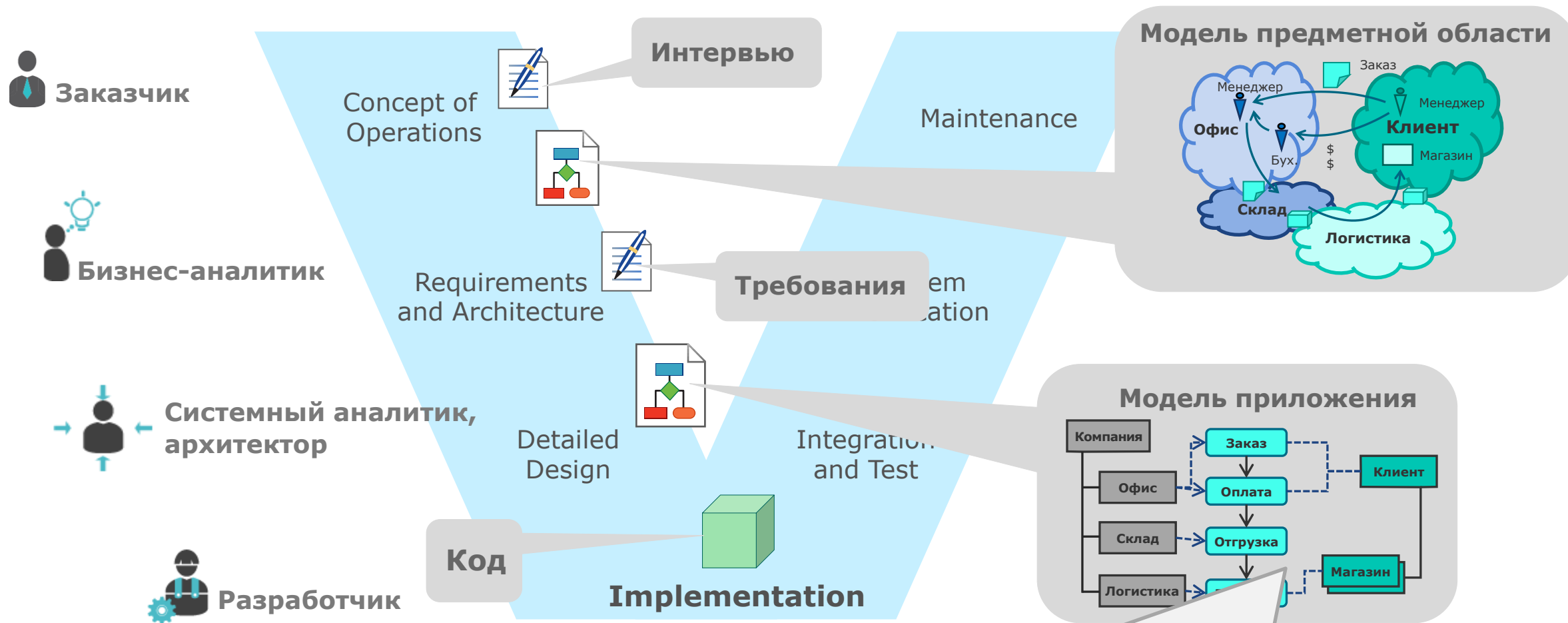
- на английском – в 2006 г.
- на русском – в 2007 г. (почти сразу!)



## Обновление от Вона Вернона

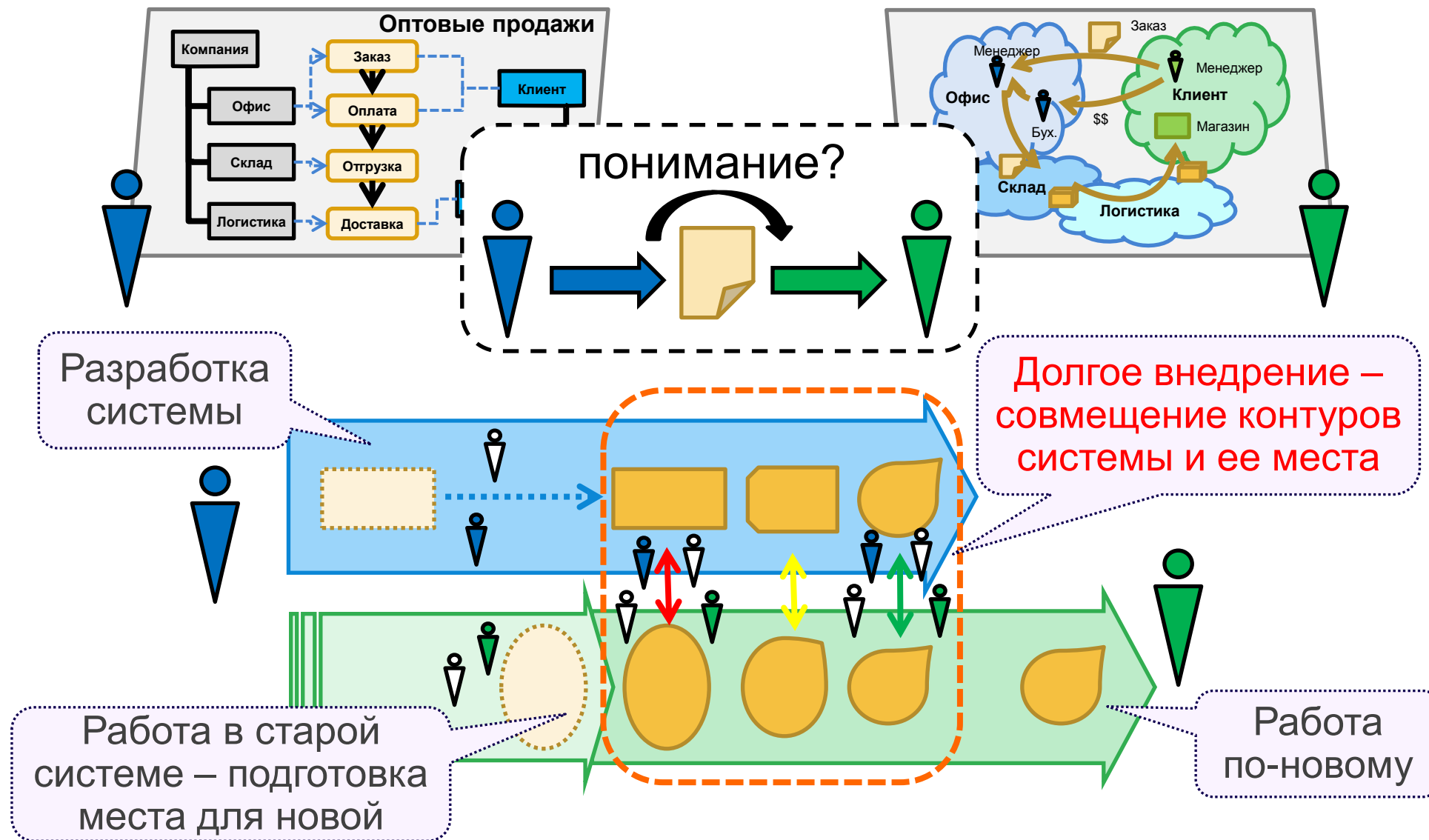
- на английском – в 2013 г.
- на русском – в 2016 г.

# Модели в классических постановках

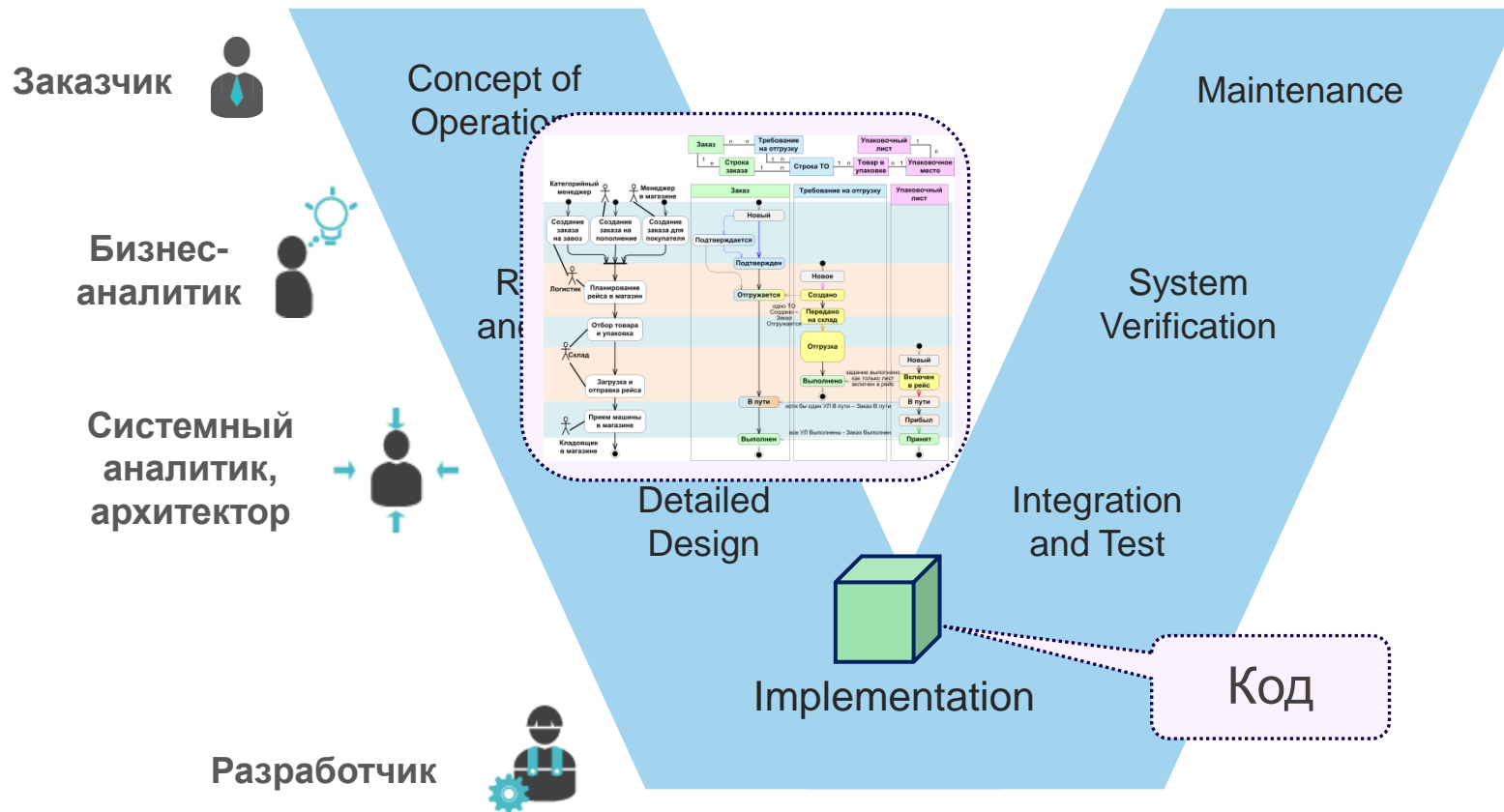


Проблема: каждый уровень отдельно – сложно изменять и поддерживать соответствие

# Проблема двух моделей



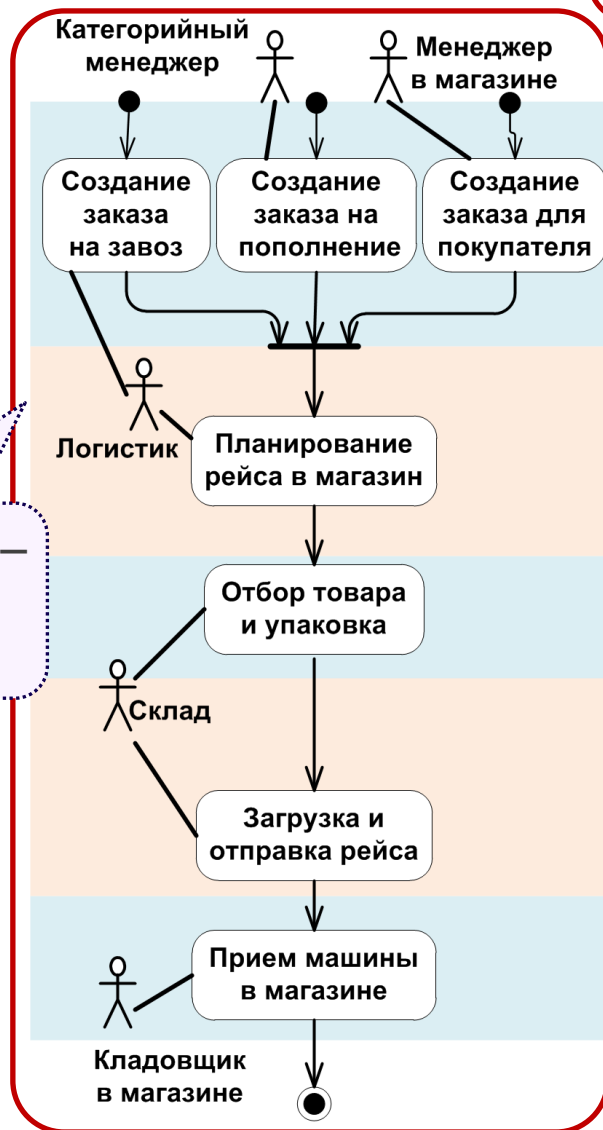
# DDD: единая модель, единый язык, прозрачное отражение в код



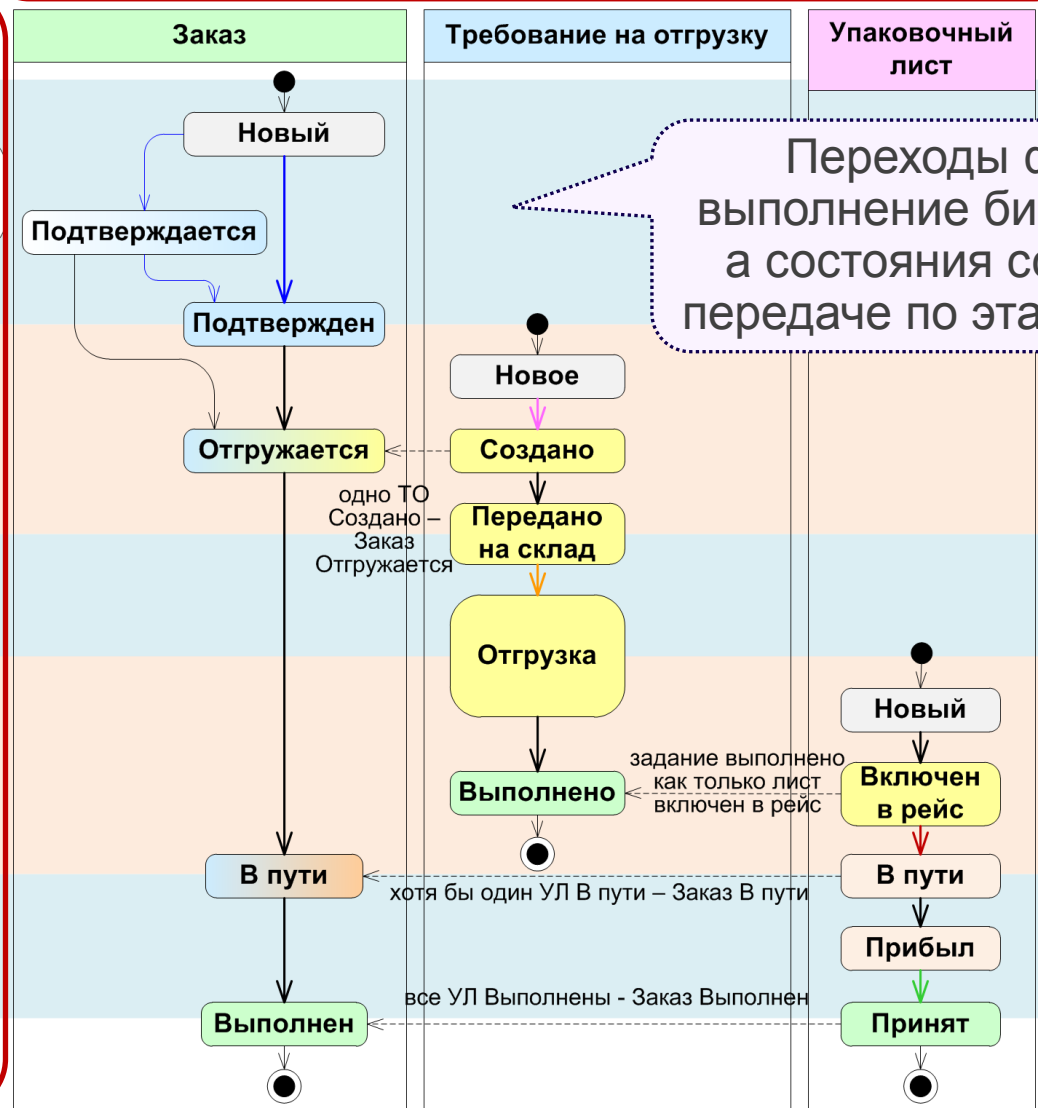
# Комплексная модель

## Снабжение магазинов

### Бизнес-процесс – диаграмма активности



Выделяем объекты-документы



Переходы фиксируют выполнение бизнес-функции, а состояния соответствуют передаче по этапам обработки

# Система дополняет ИТ-ландшафт

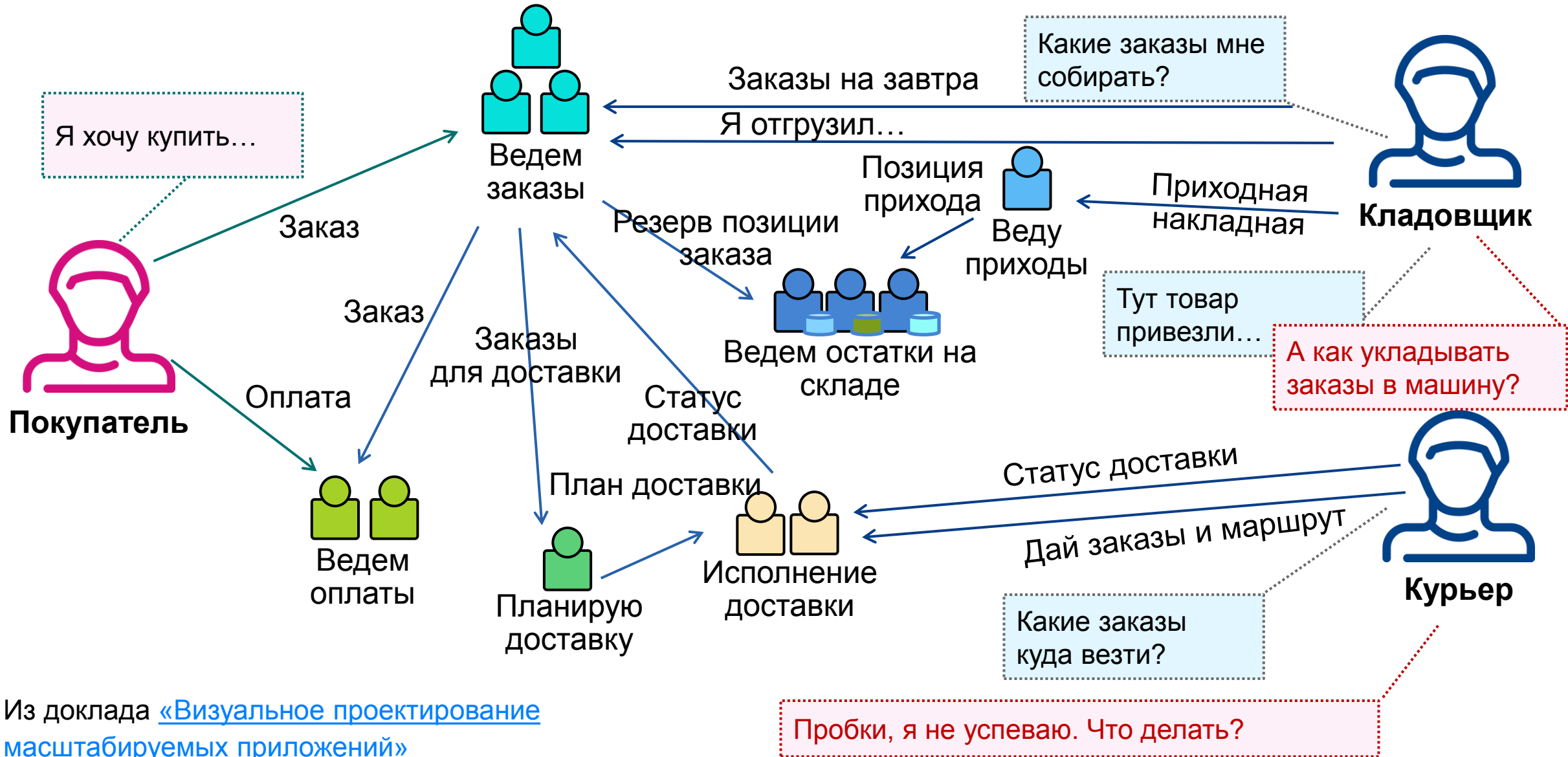




# Система становится основой

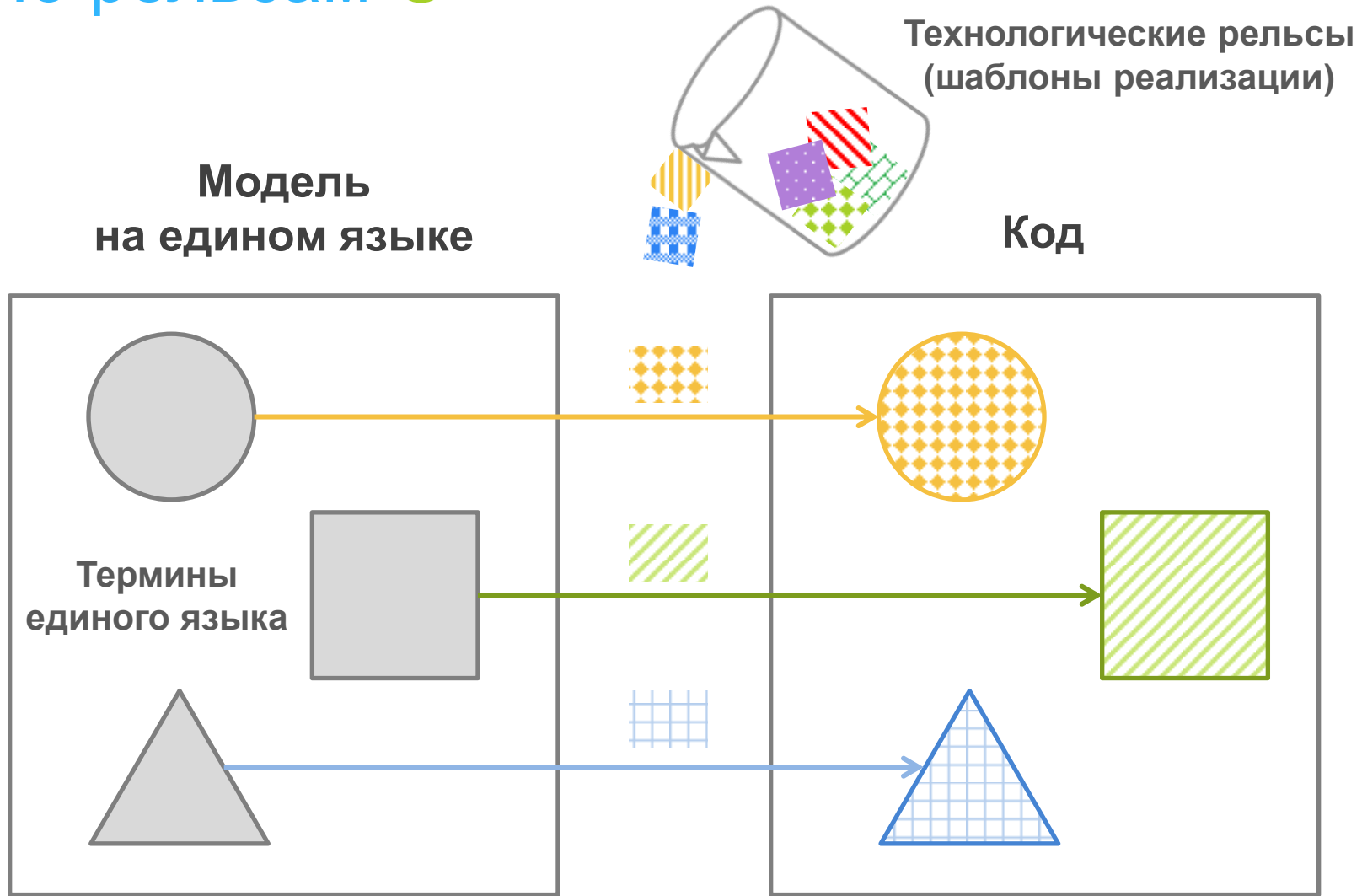


# DDD и сервисная архитектура приложения



Из доклада [«Визуальное проектирование масштабируемых приложений»](#)

# По рельсам 😊



# Шаблоны работы с контекстами



# Решение – ограниченный контекст

