

Визуальные модели корпоративного приложения

Максим Цепков

Главный архитектор решений [CUSTIS](#)

<http://mtsepkov.org>

Москва, 1 ноября 2017

Корпоративное приложение – часть архитектуры предприятия

Для проектирования используем визуальные модели:

- ➔ Архитектура предприятия в целом и бизнес-уровень
- ➔ Метафора системы – выражение системы в целом
- ➔ DDD для корпоративного приложения –
шаблон Учетной машины и визуальные представления

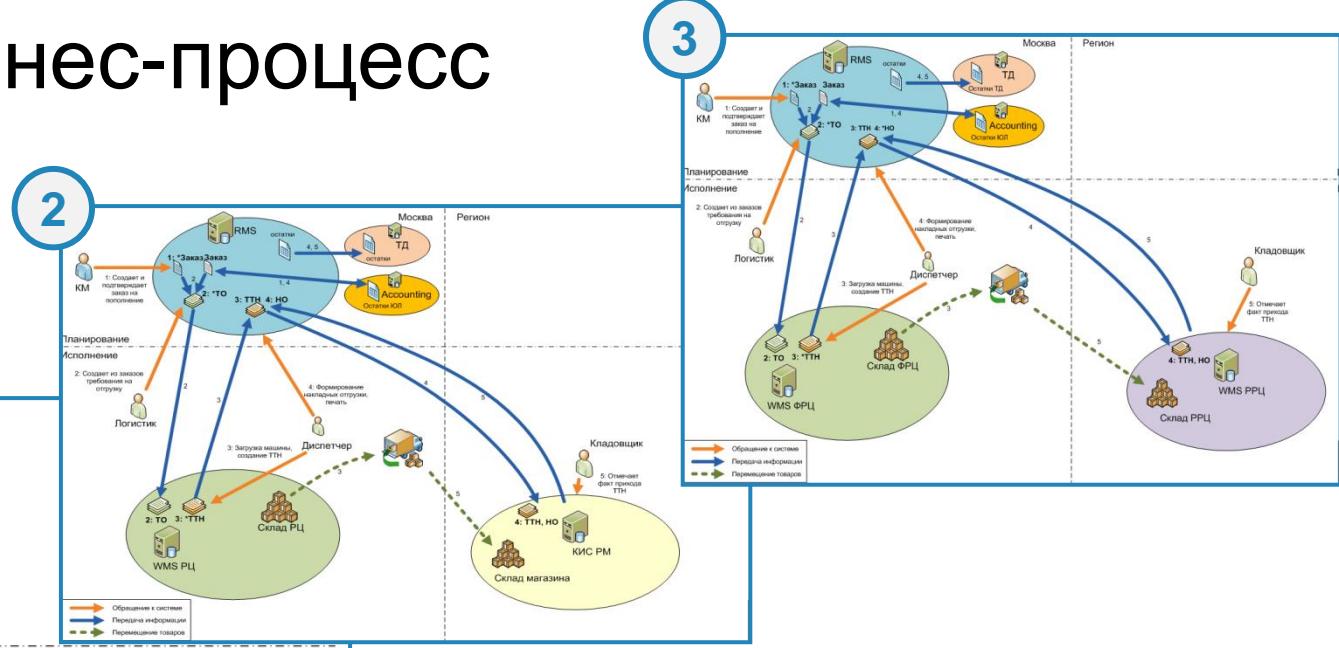
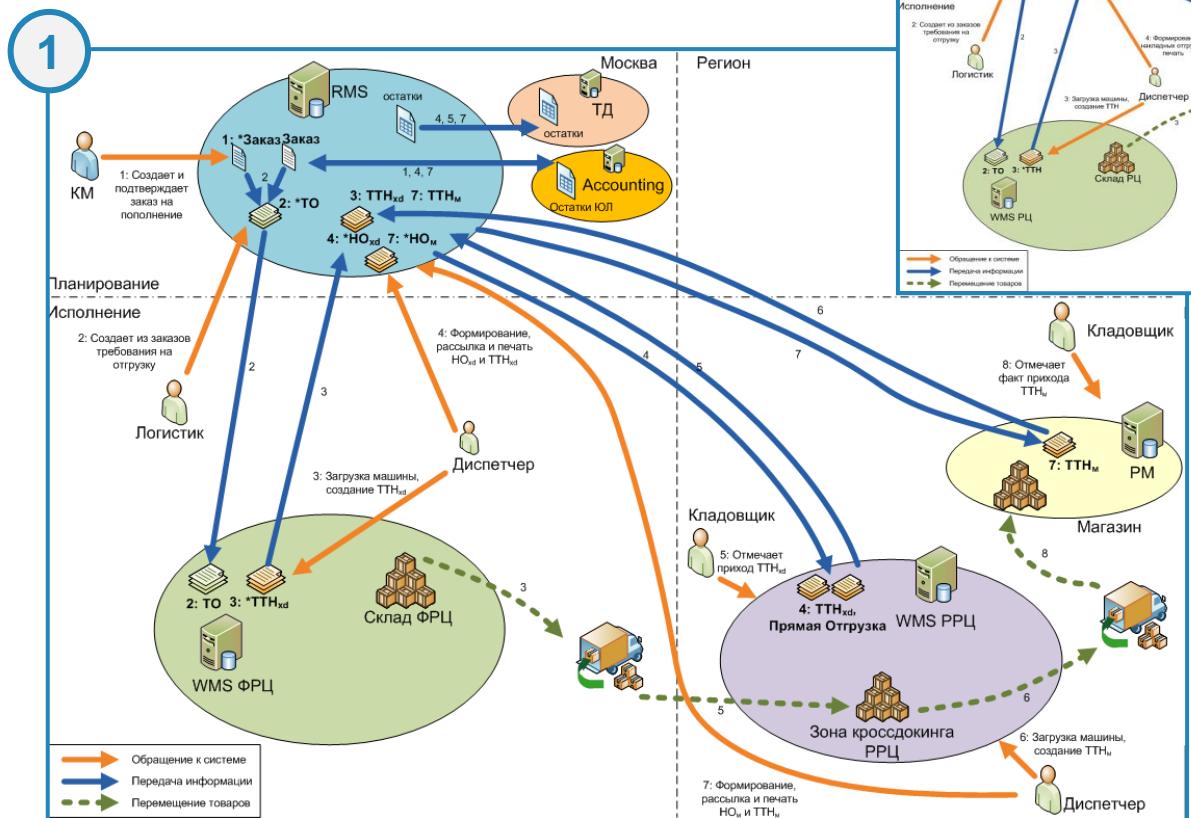
Archimate: концептуальная модель



Архитектура предприятия в целом и бизнес-уровень

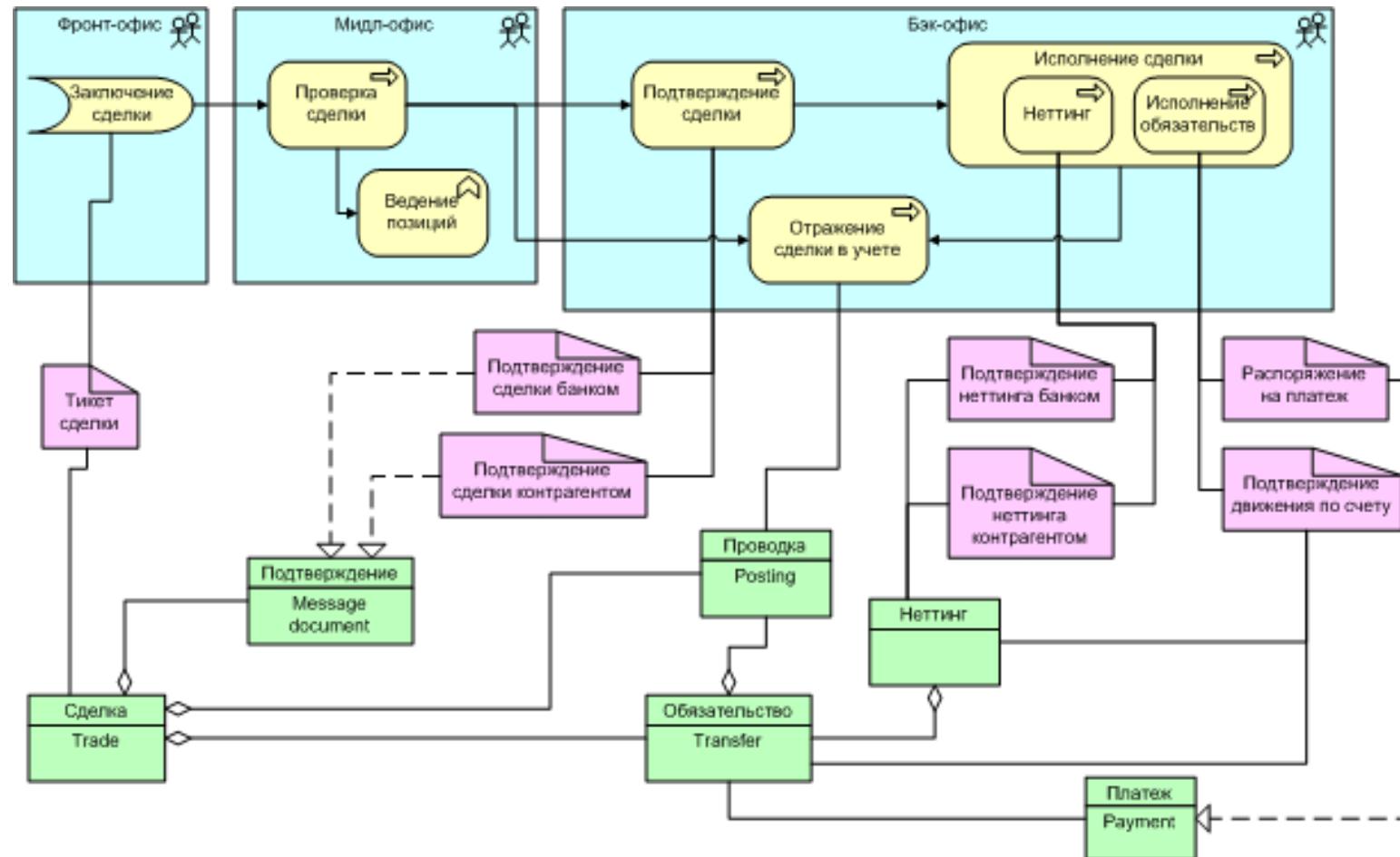
Неформальный бизнес-процесс

Пример:
снабжение магазинов

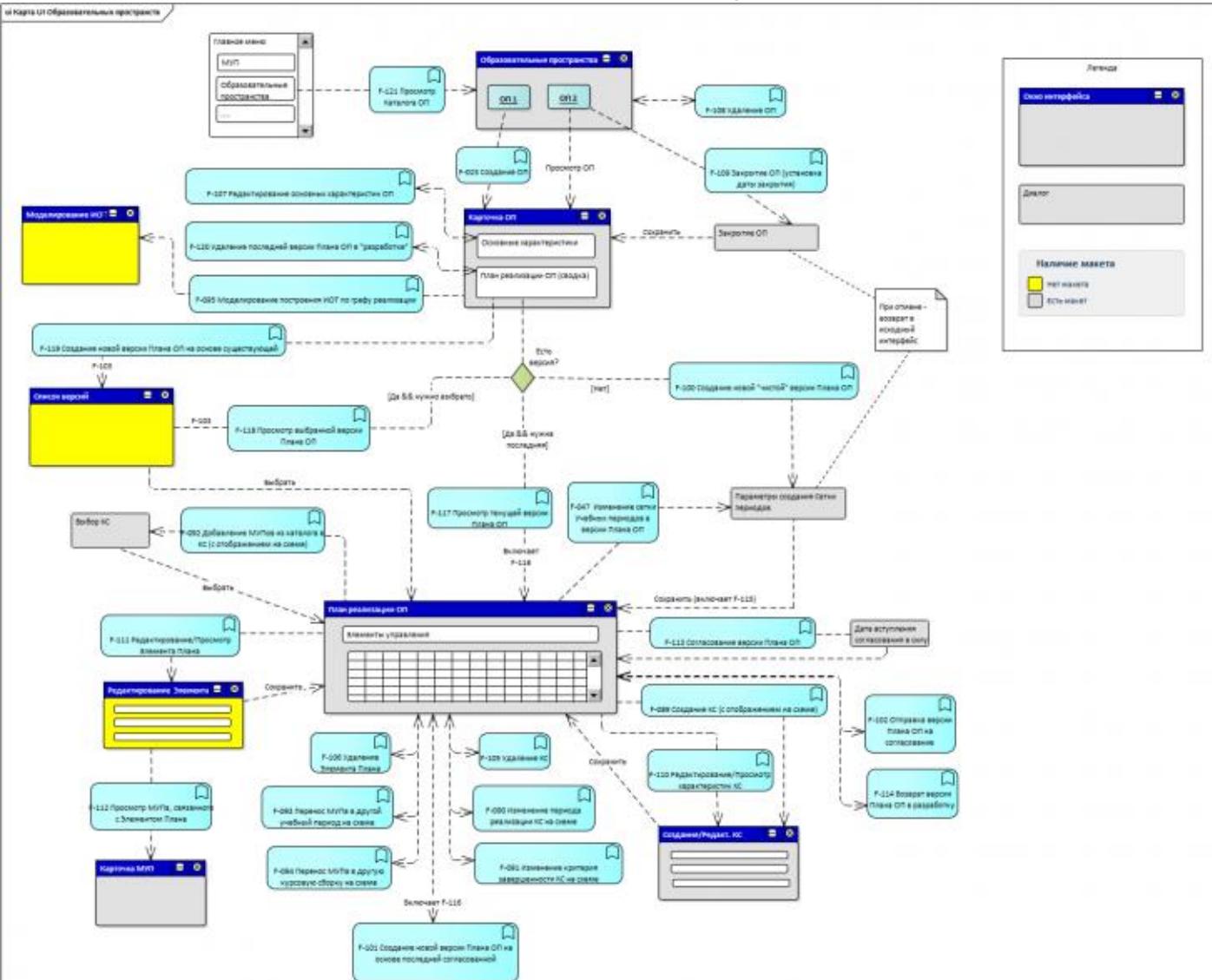


Комплексная схема
3 уровней Archimate,
но в вольной нотации
она понимается лучше

Связь бизнес-процессов и объектов приложения

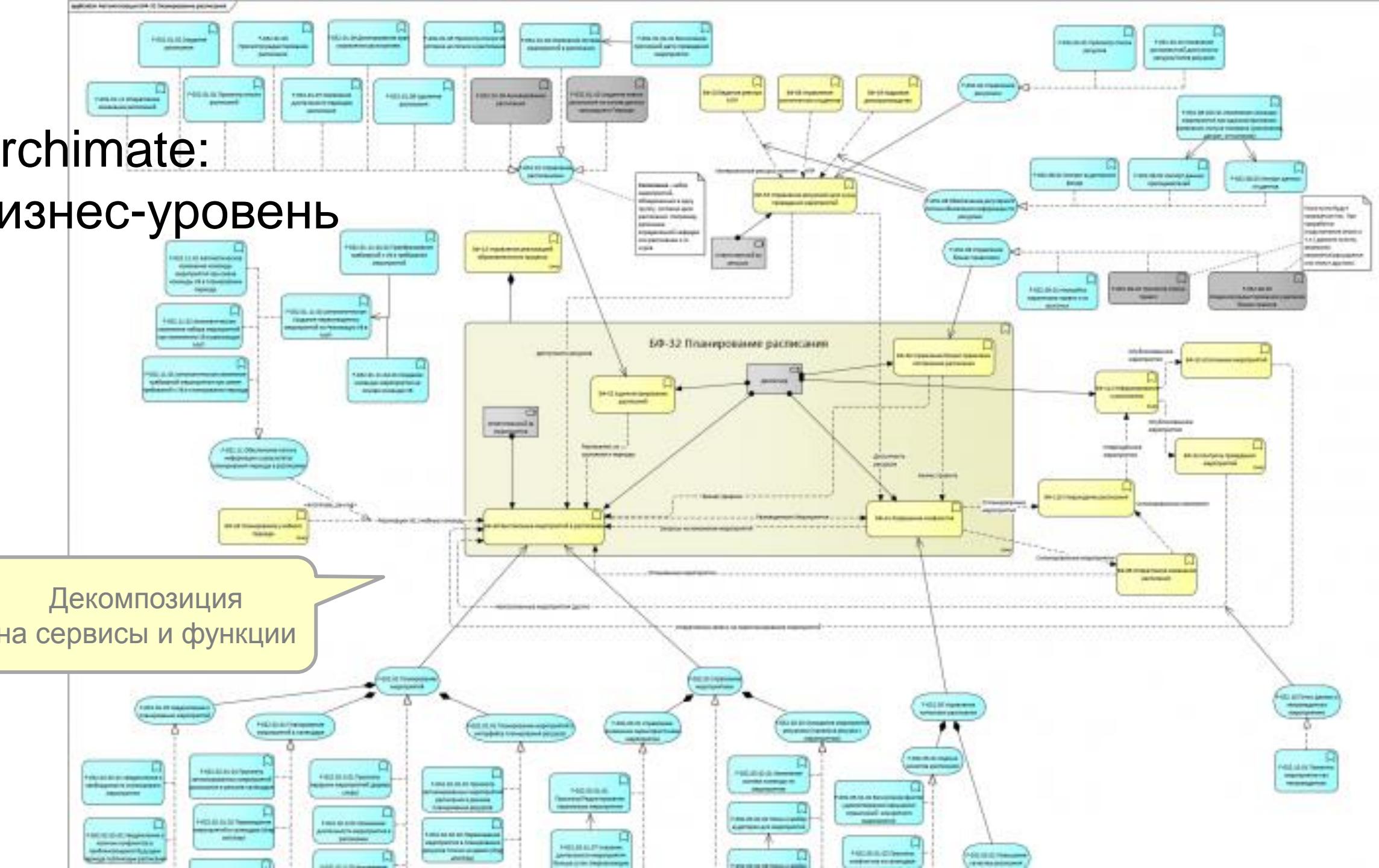


Поддержка бизнес-функций экранами приложения



Archimate: бизнес-уровень

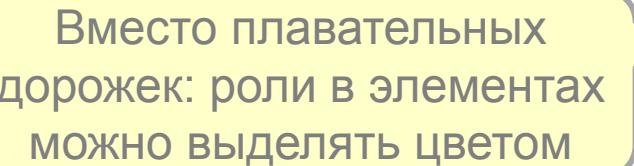
Декомпозиция
на сервисы и функции



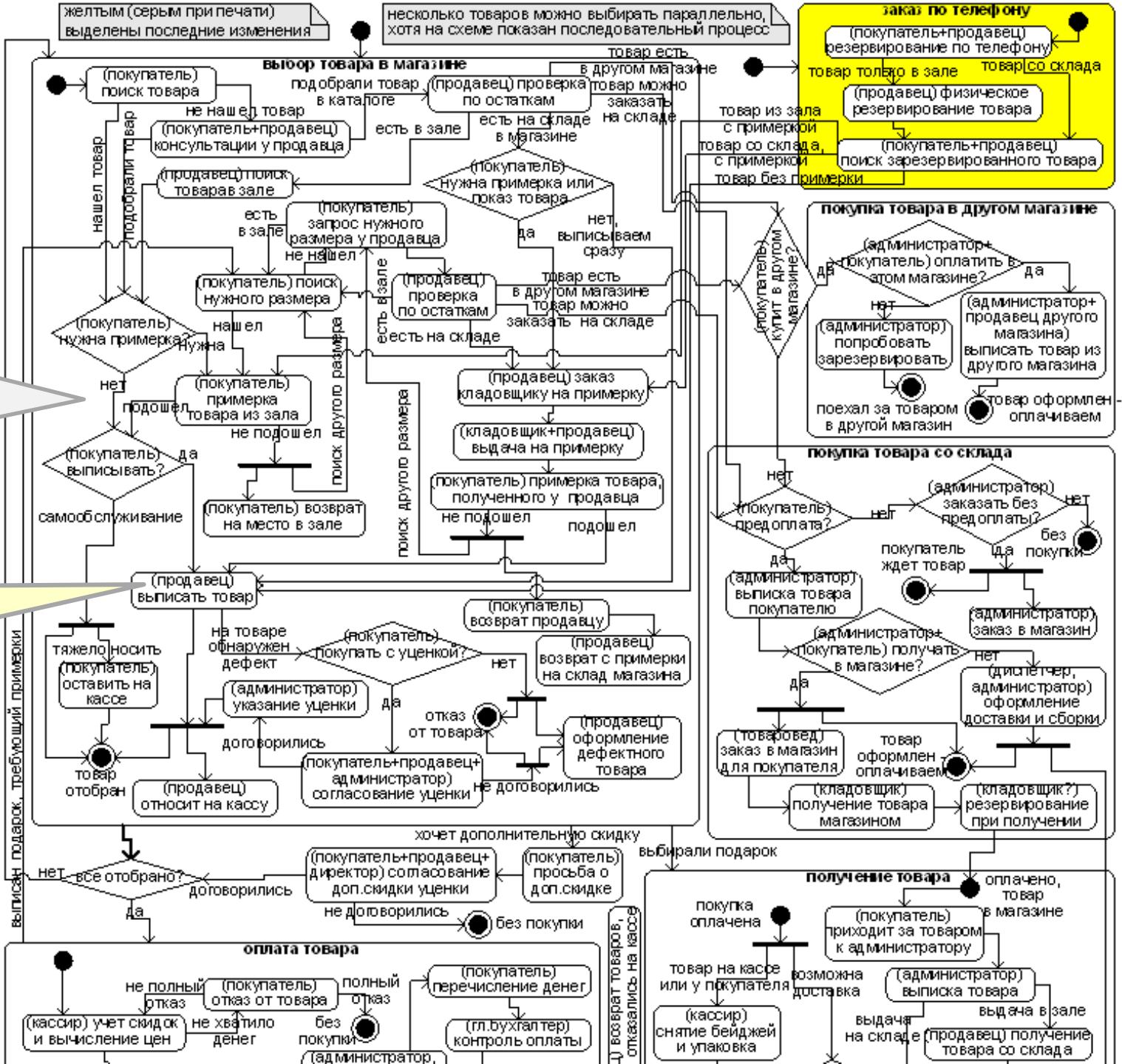
Диаграммы деятельности



Даже в «очевидном» процессе обслуживания покупателей выясняешь многое особенностей



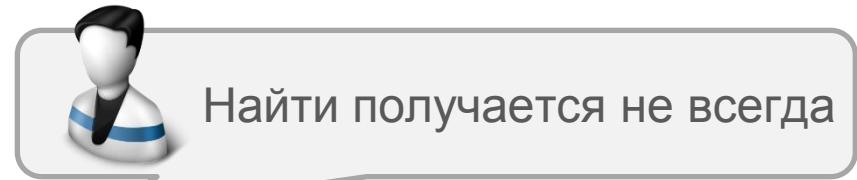
Заказчик проверяет
детали лучше,
чем по тексту



Метафора системы

Что такое метафора системы

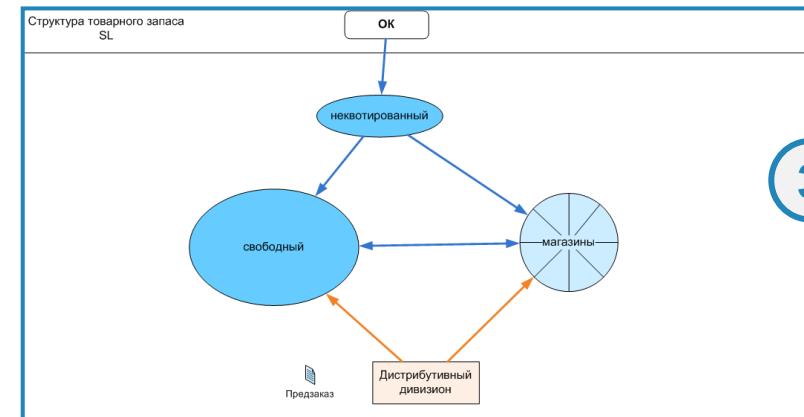
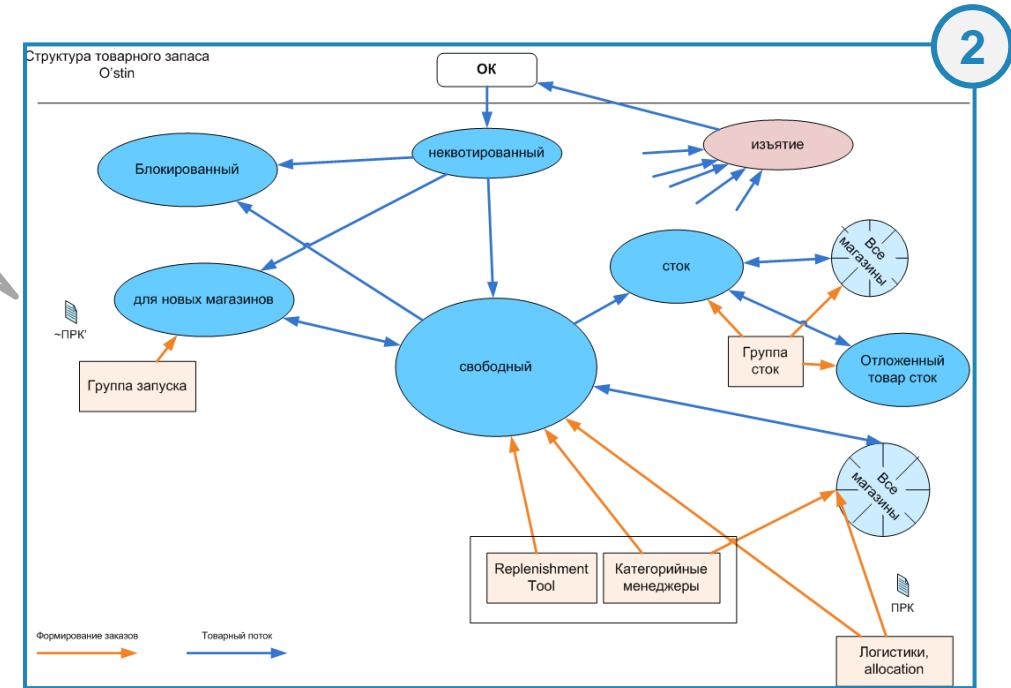
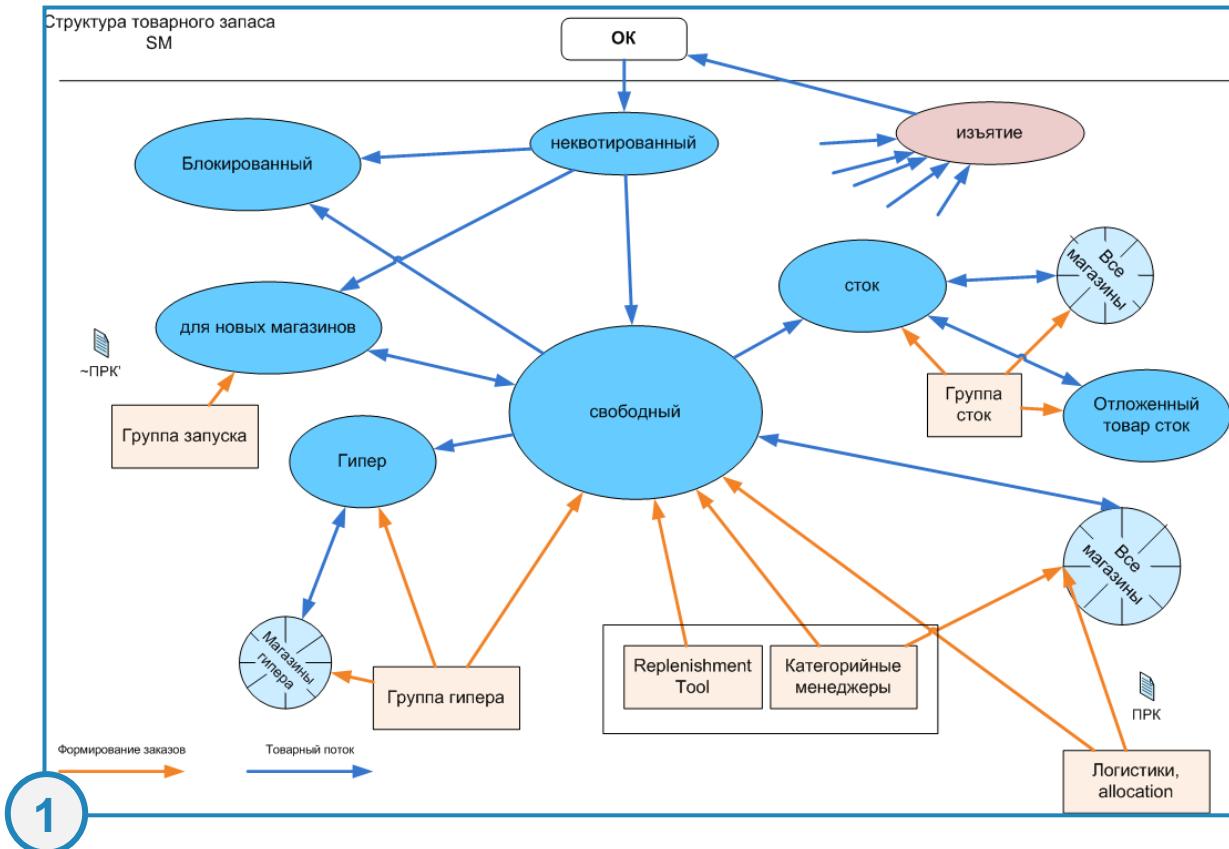
- ➡ Метафора – лаконичная концепция, которая много говорит о системе
- ➡ Иными словами, проекция (viewpoint), максимально отражающая суть
- ➡ Визуальный образ – хорошая метафора
 - Дает цельное представление о системе
 - Обеспечивает взаимопонимание
- ➡ Поиск метафоры начинается с vision-проекта
- ➡ Визуальная основа – стандартна или придумана



Используем типовые образы

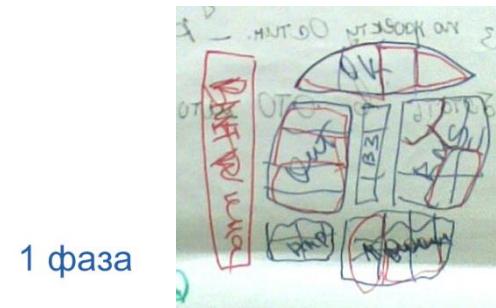
Пример: деление товара

Варианты процессов в простой нотации, схемы понимаются на лету

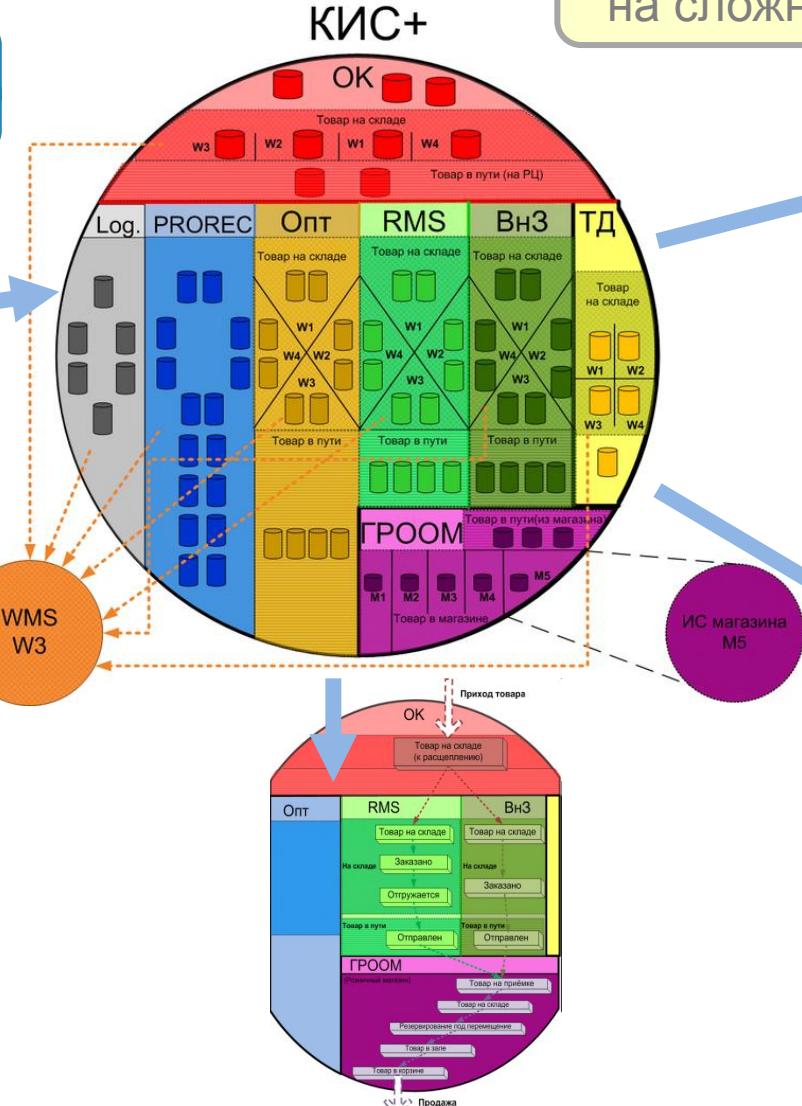
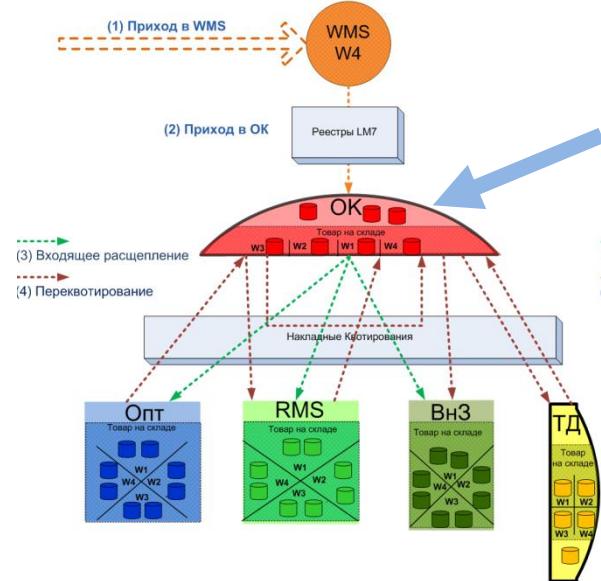


Придумываем свой образ

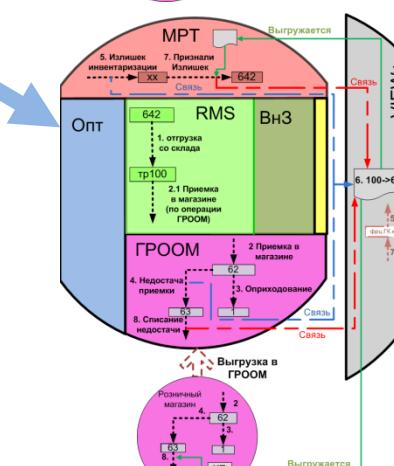
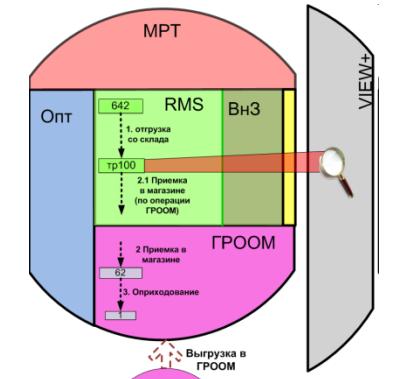
Пример: единая витрина
для многих систем



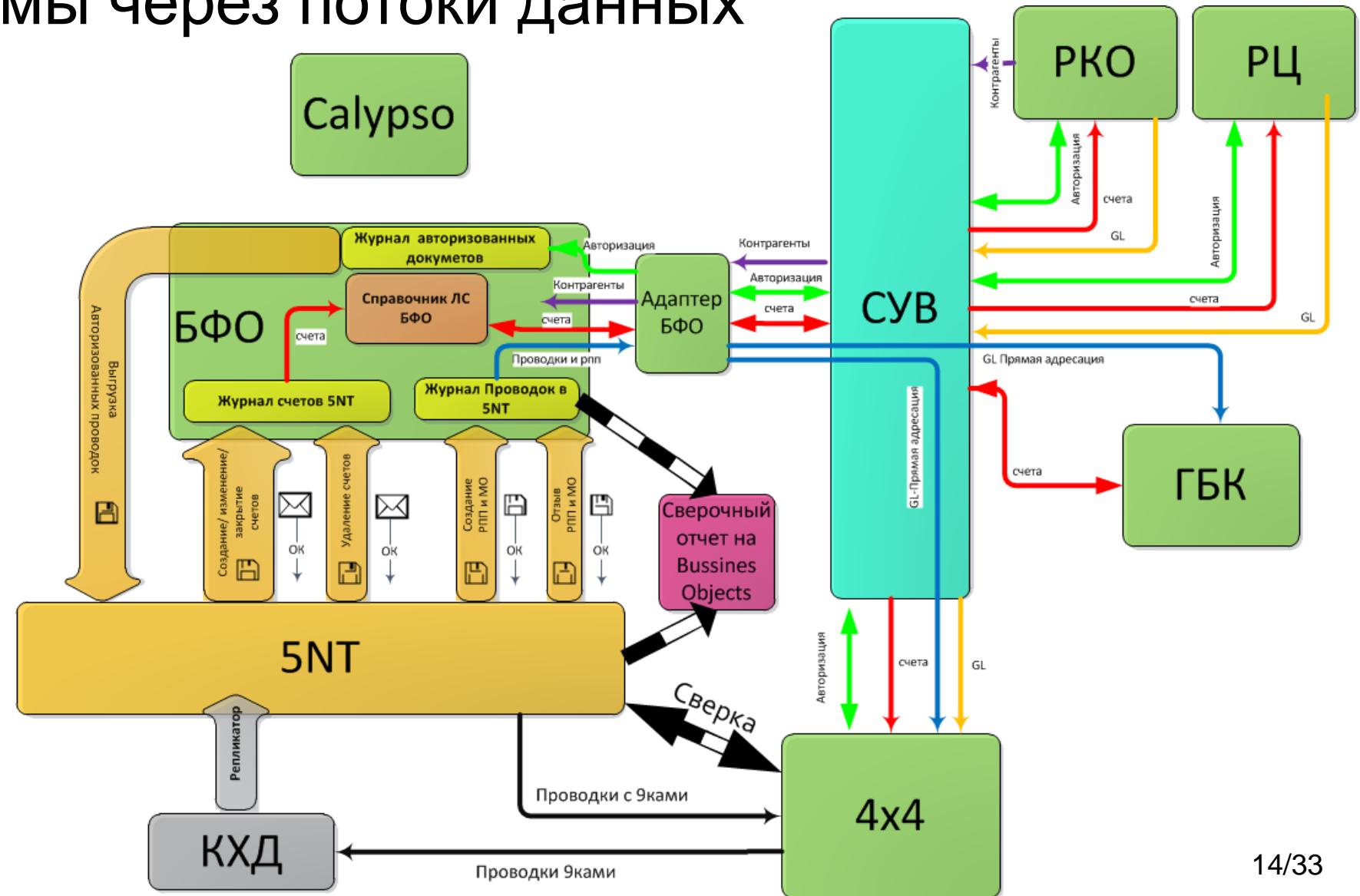
1 фаза

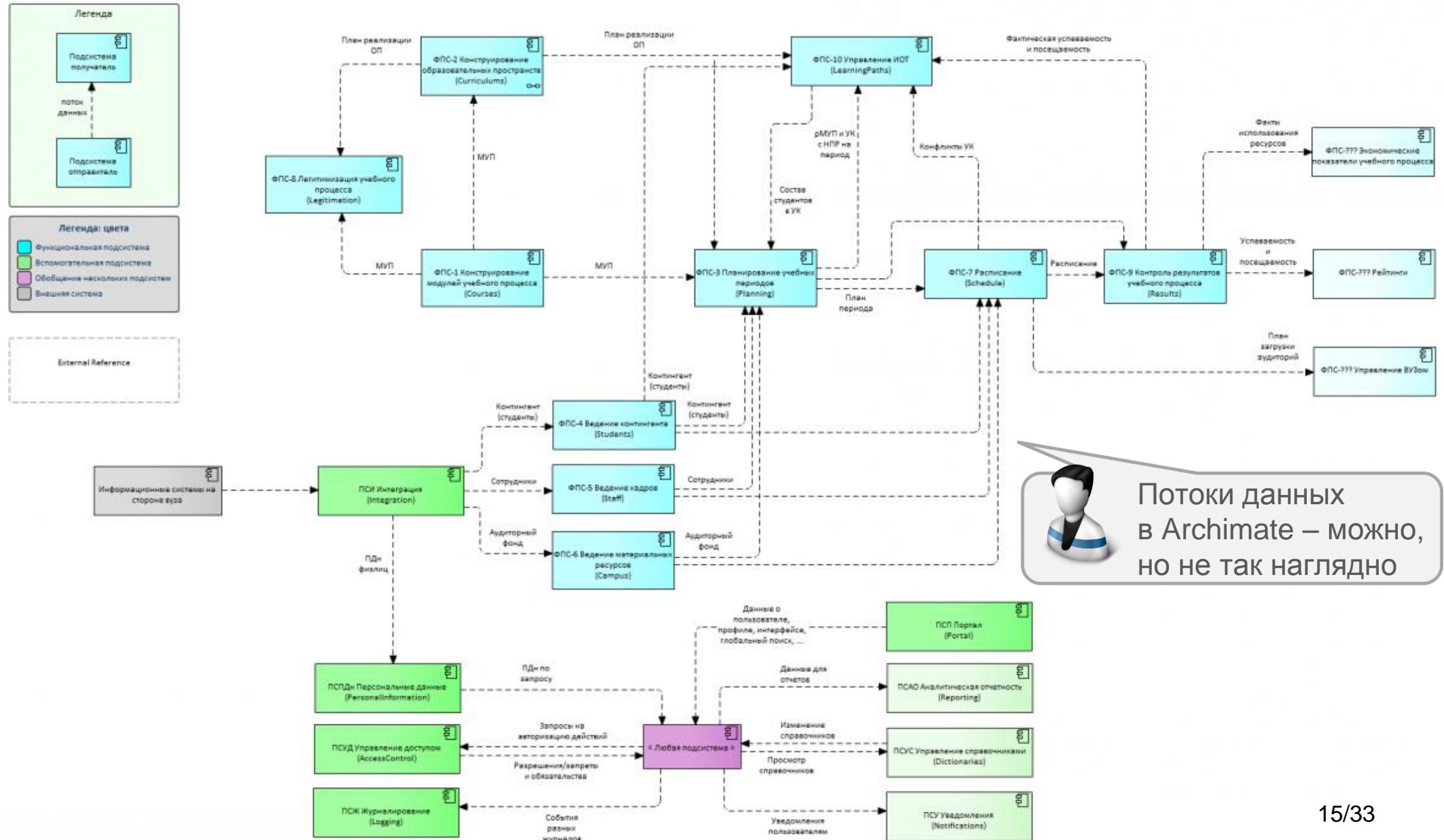


Лупа: смотрим
на сложную структуру



Роль системы через потоки данных

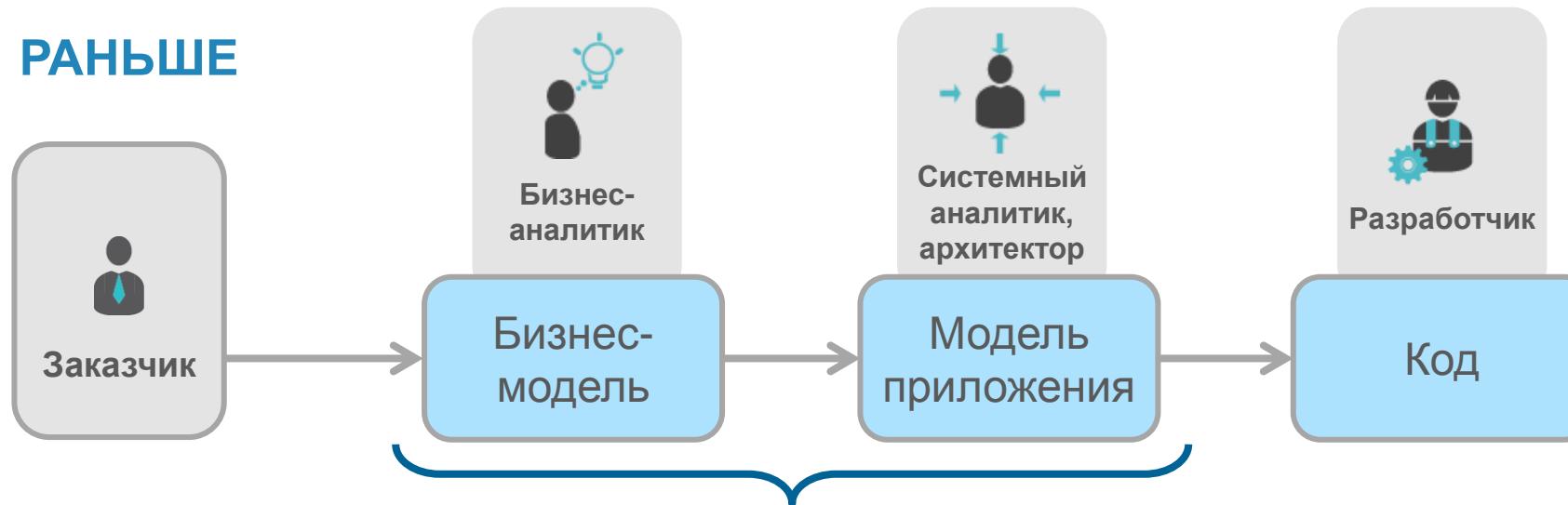




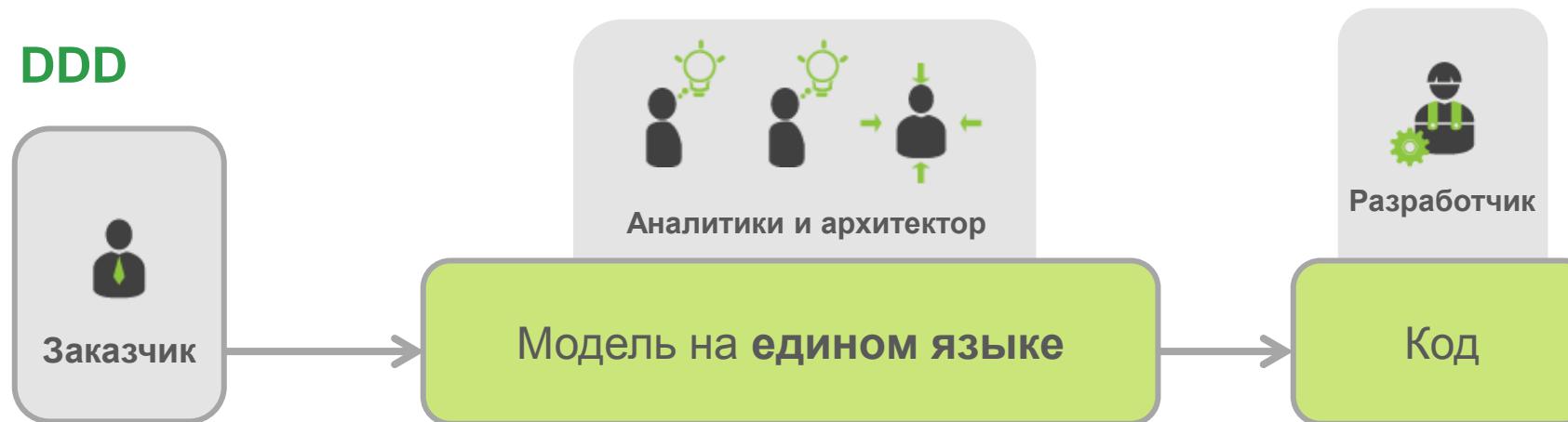
DDD для корпоративного приложения – шаблон Учетной машины

Основная идея DDD

РАНЬШЕ



DDD



Единый язык (ubiquitous language)

- ➡ Построен на основе терминов предметной области
- ➡ Его понимают ИТ-специалисты и эксперты бизнеса
- ➡ На нем описана модель ИТ-системы и ее место в бизнес-процессах



Понятия единого языка: **клиент, накладная, платеж, долг**, – из предметной области

Единая модель

- ➡ Аналитик собирает требования и строит модель: сначала – предметной области, затем – системы
- ➡ Артефакты модели описывают и систему, и ее использование в бизнес-процессах предприятия
- ➡ Разработчик реализует модель
- ➡ Артефакты модели можно проследить в коде



Модель предметной области становится моделью системы

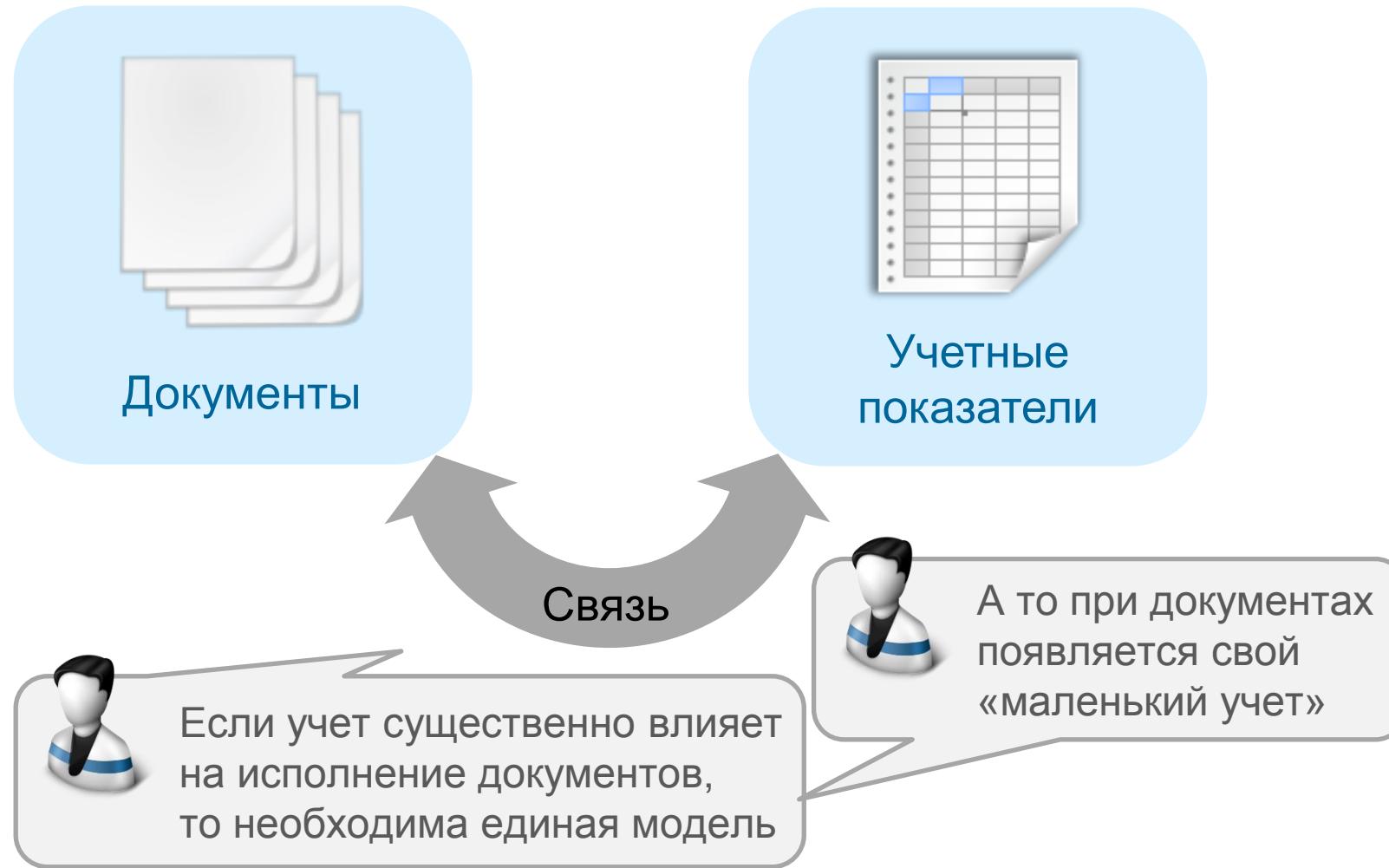
Плюсы единой модели

- 😊 Верификация постановок бизнес-специалистами
- 😊 Общее понимание требований на стороне бизнеса
- 😊 Обсуждение модели бизнесом и ИТ, поиск баланса в сложных решениях
- 😊 Перенос моделей из других предметных областей
- 😊 Бизнес представляет потенциальные возможности системы и сложность различных доработок
- 😊 На этапе эксплуатации – эффективное общение бизнес-пользователей и ИТ без квалифицированных переводчиков-аналитиков

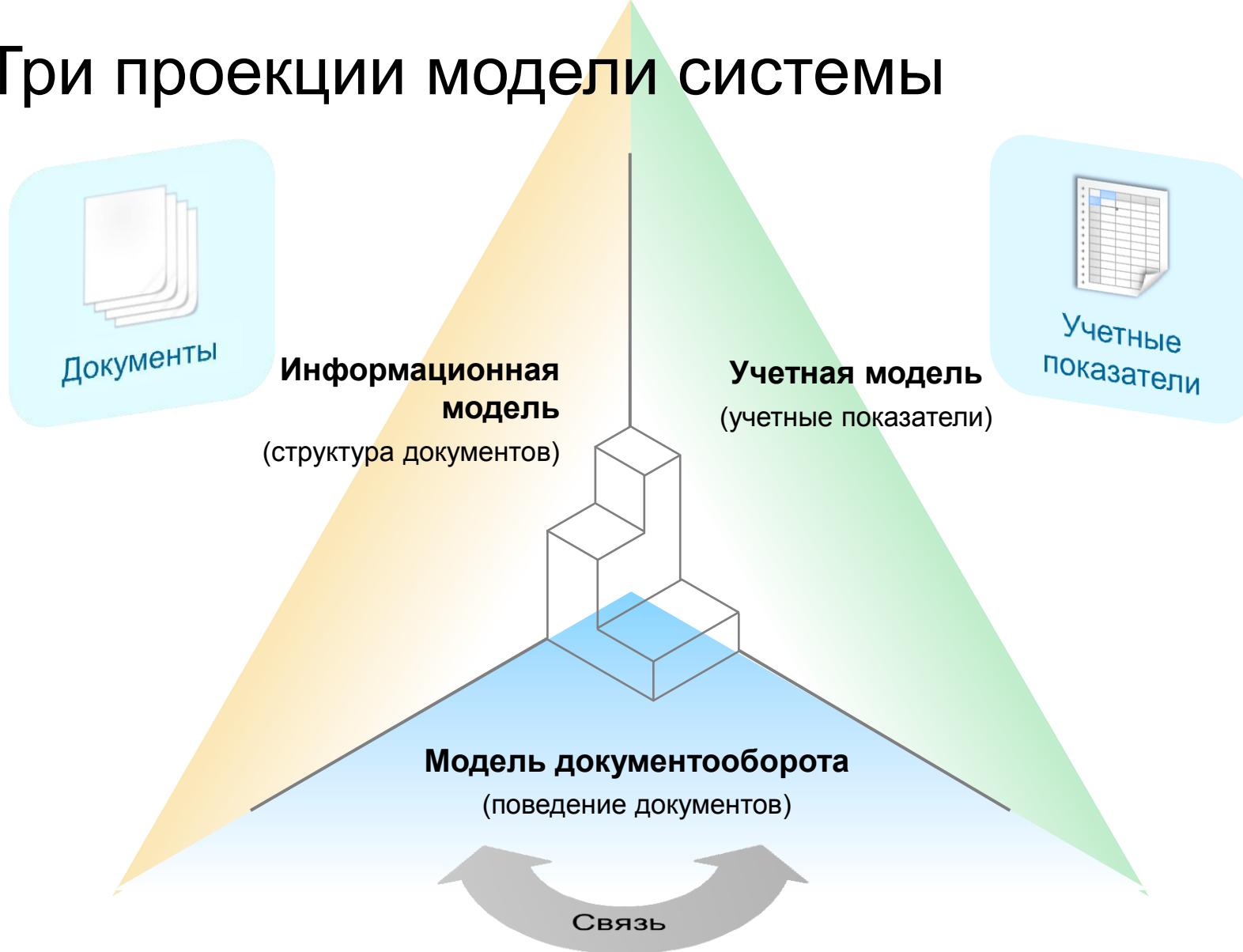
Типичное корпоративное приложение

- ➡ Пользователи создают документы  **Объектная модель**
- ➡ По необходимости заполняют справочники
- ➡ Потом документы исполняют  **Жизненный цикл документа**
- ➡ При этом меняются учетные данные
- ➡ Которые влияют на исполнение документов  **Учет –
не объектная модель**
- ➡ И отражаются в отчетах

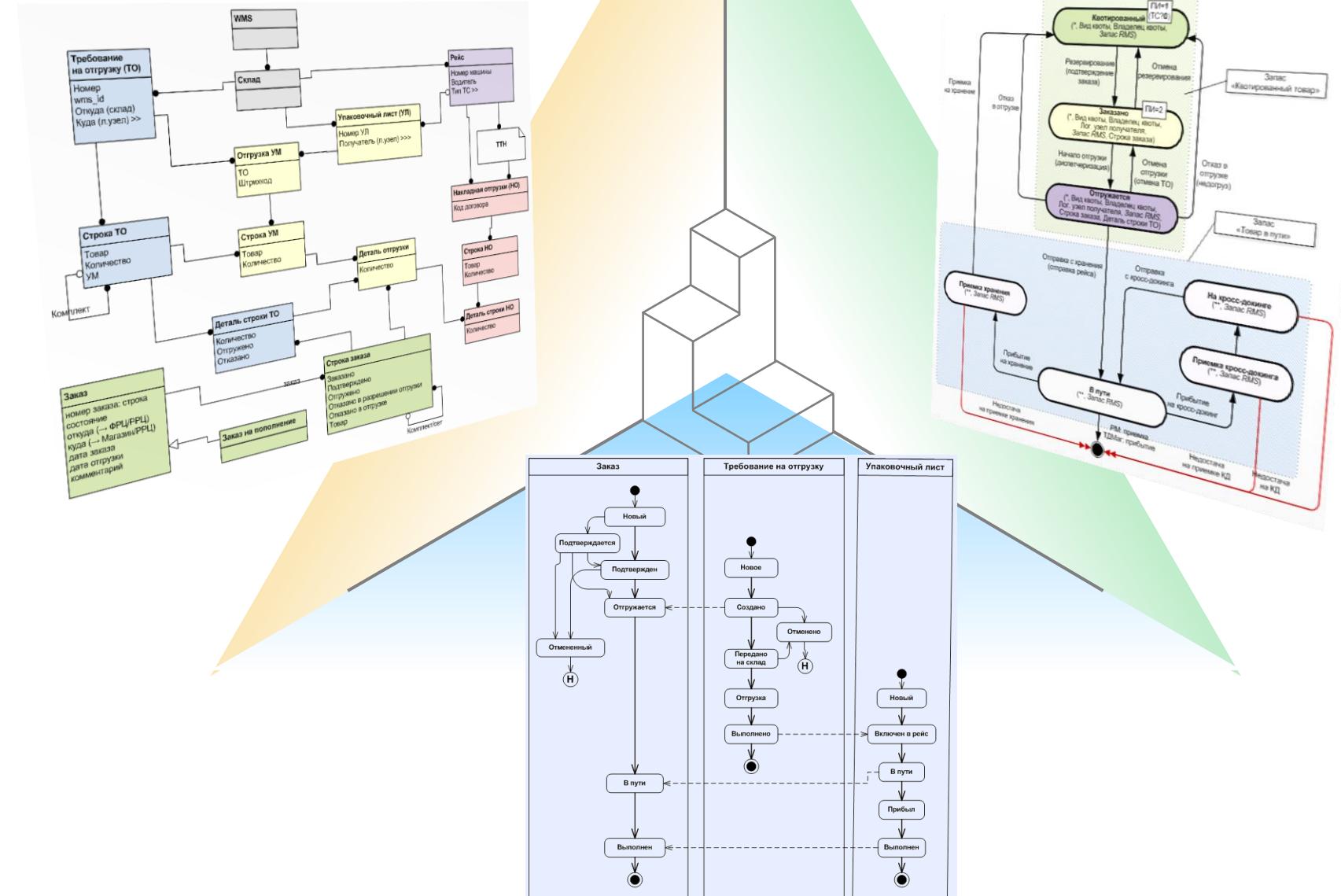
Одна модель или несколько?



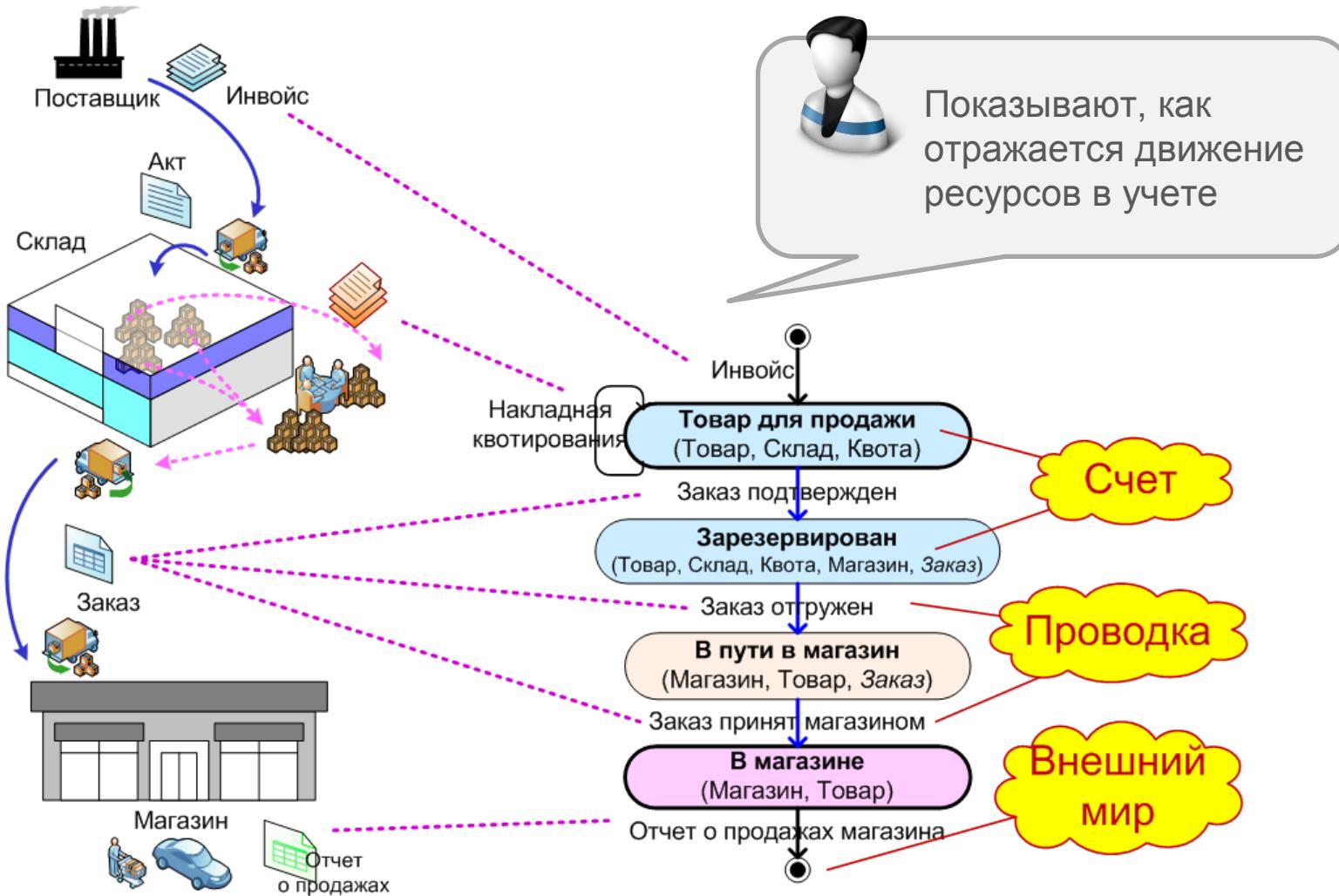
Три проекции модели системы



Диаграммы для проекций



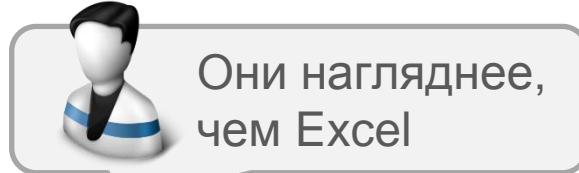
Диаграммы учета



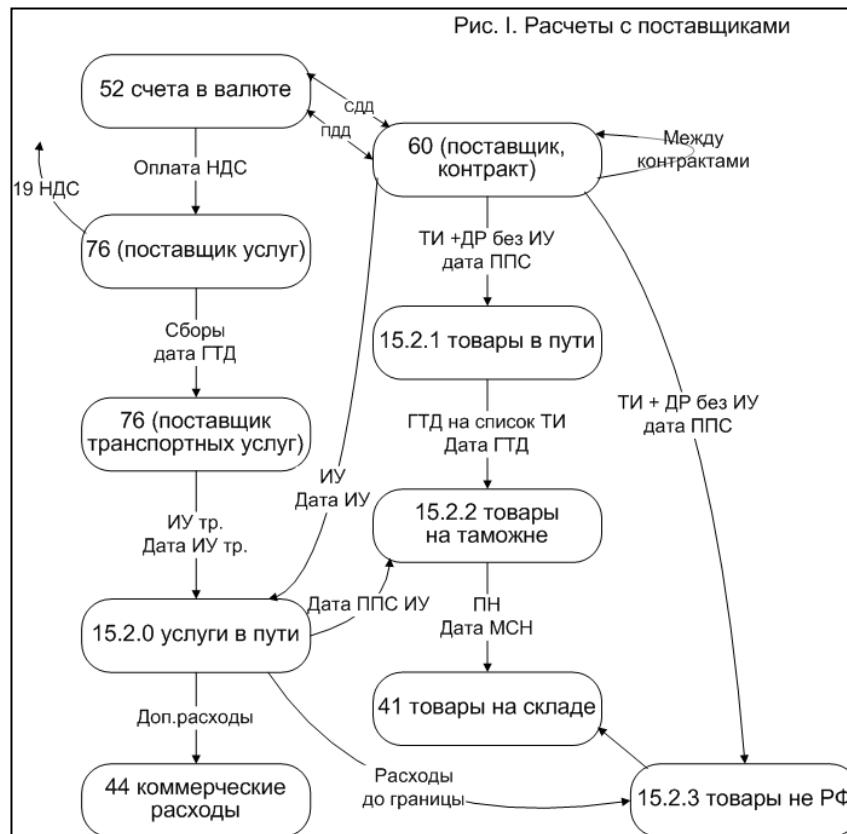
Статья [«Диаграммы учета: мост между бухгалтером и разработчиком»](#)
(журнал «Бухгалтер и компьютер», №5-2011)

Диаграммы учета для бизнеса

Бухгалтеры могут применять диаграммы учета для разработки учетной политики.



Они нагляднее,
чем Excel



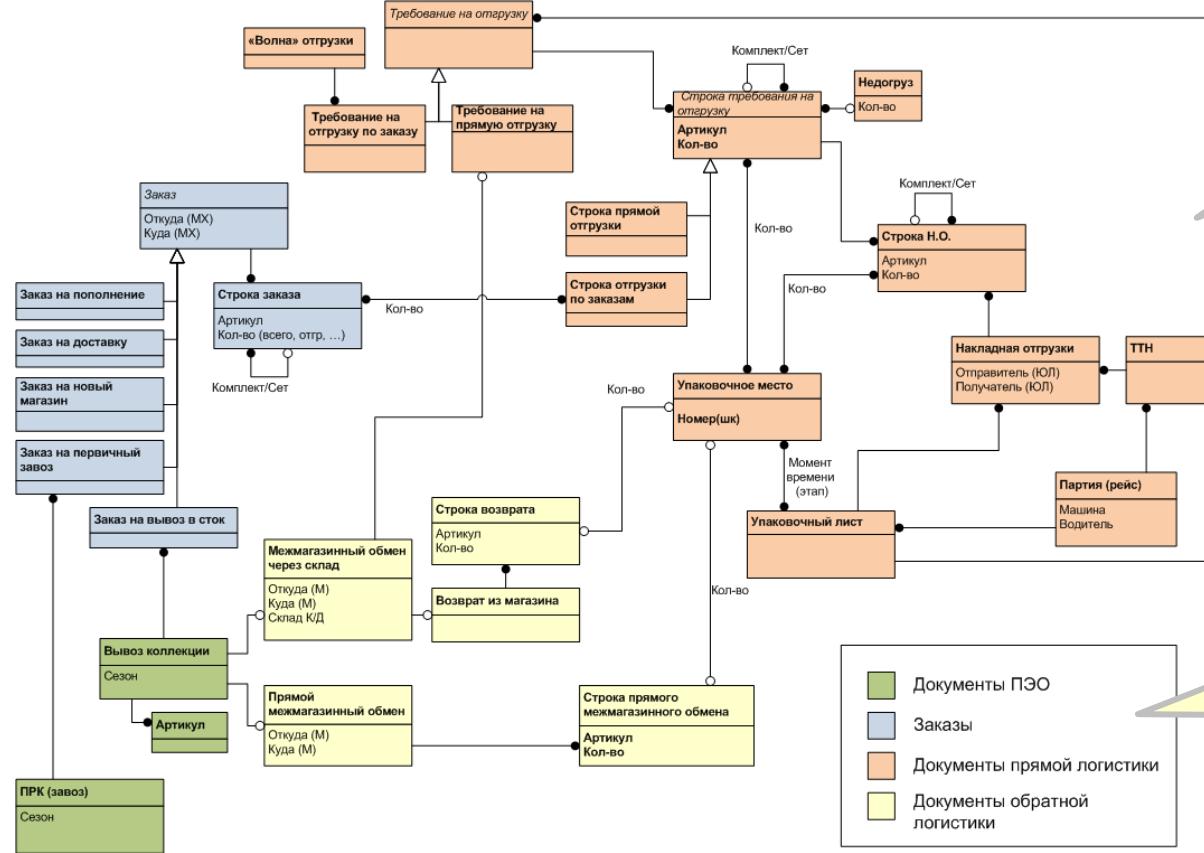
Номер записи	Начало отчетного периода	Номер записи	Начало отчетного периода	УЧЕТ ОБРАЗЦЫ С ТОВАРЫМИ			Текущий баланс	Приращение суммы продаек	Без НДС	Выручка без дополн.
				Баланс	Приращение	Баланс				
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
Приращение - пополнение дебитора										
22										
23										
24										
25										
26										
27										
28										
29										
30										
Приращение - пополнение в кредит										
31										
32										
33										
34										
35										
36										
37										
38										
39										
40										
41										
42										
43										
44										
45										
46										
47										
48										
49										
50										
51										
52										
53										
54										
55										
56										
57										
58										
59										
60										
61										
62										
63										
64										
65										
66										
67										
68										
69										
70										
71										
72										
73										
74										
75										
76										
77										
78										
79										
80										
81										
82										
83										
84										
85										
86										
87										
88										
89										
90										
91										
92										
93										
94										
95										
96										
97										
98										
99										
100										
101										
102										
103										
104										
105										
106										
107										
108										
109										
110										
111										
112										
113										
114										
115										
116										
117										
118										
119										
120										
121										
122										
123										
124										
125										
126										
127										
128										
129										
130										
131										
132										
133										
134										
135										
136										
137										
138										
139										
140										
141										
142										
143										
144										
145										
146										
147										
148										
149										
150										
151										
152										
153										
154										
155										
156										
157										
158										
159										
160										
161										
162										
163										
164										
165										
166										
167										
168										
169										
170										
171										
172										
173										
174										
175										
176										
177										
178										
179										
180										
181										
182										
183										
184										
185										
186										
187										
188										
189										
190										
191										
192										
193										
194										
195										
196										
197										
198										
199										
200										
201										
202										
203										
204										
205										
206										
207										
208										
209										
210										
211										
212										
213										
214										
215										
216										
217										
218										
219										
220										
221										

Документооборот – объектная модель

- ▶ Объекты с атрибутами и методами – элементы единого языка для предметной области и способ их соединения в сложные конструкции
- ▶ Для отражения документооборота используем **состояние документа**
- ▶ **Диаграмма классов и диаграмма состояний UML** – визуальный образ для наглядного представления
- ▶ Объекты в программе – способ отражения модели в реализацию

Модель должен понимать заказчик

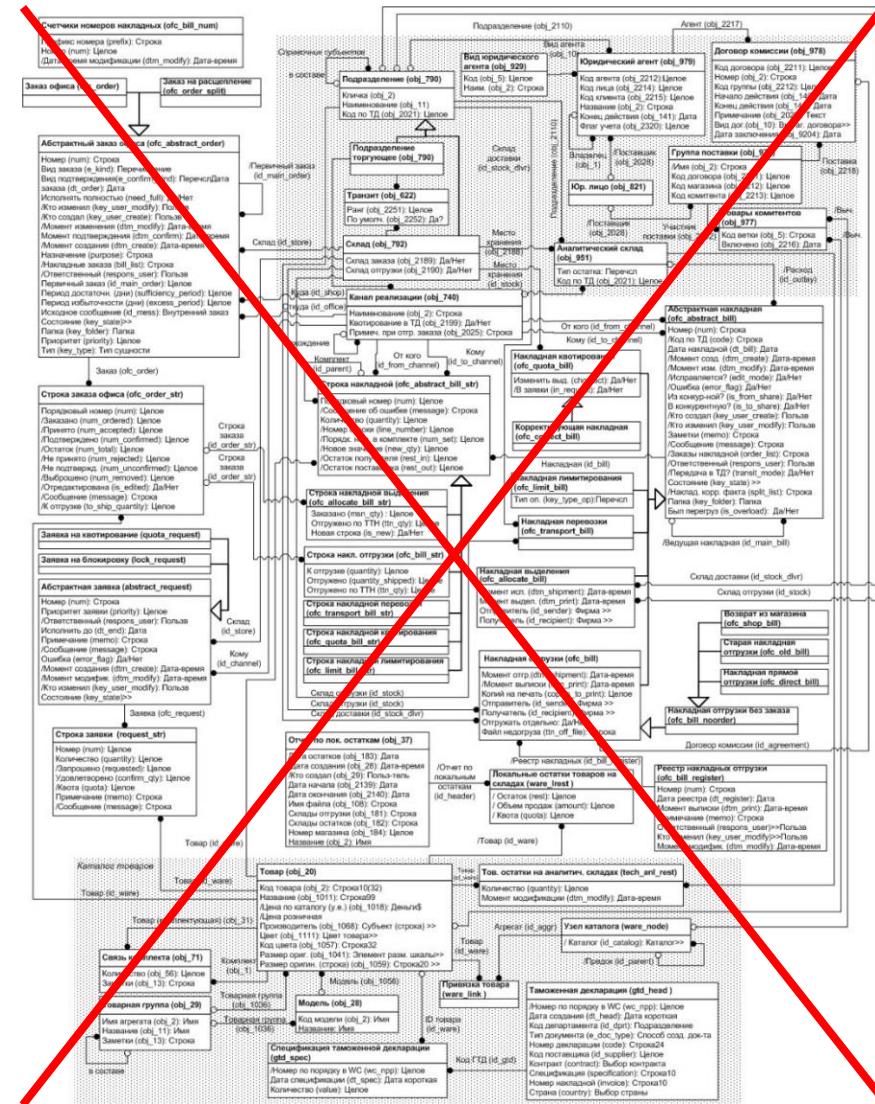
Классы соответствуют бизнес-объектам –
заказчик видит знакомые названия



Представляем диаграммой классов

Используем цветовые выделения

Не нужно усложнять диаграмму



 Не нужно

- Ставить изобразить все классы на одной диаграмме
 - Отображать вспомогательные атрибуты
 - Использовать технические коды
 - Применять сложную нотацию

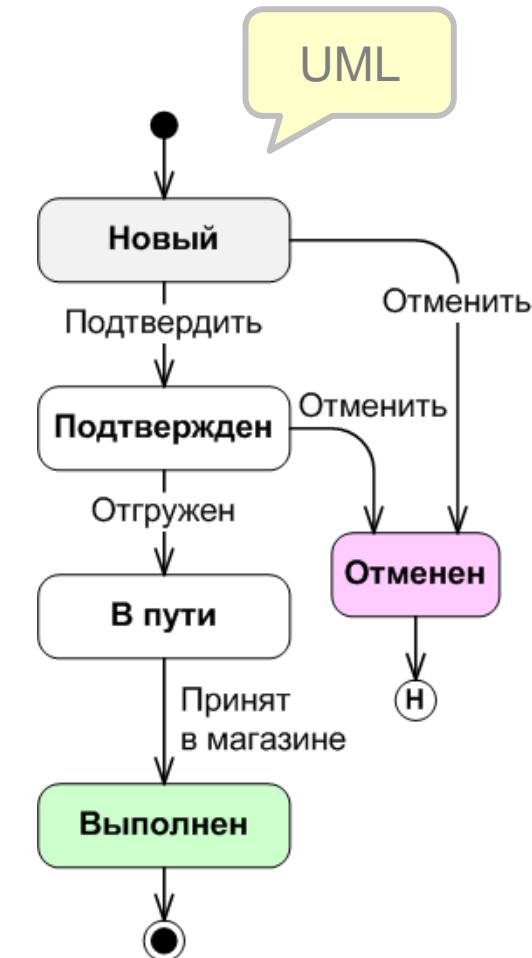
😊 Диаграммы должны понимать заказчики, а не только разработчики

Модель документооборота

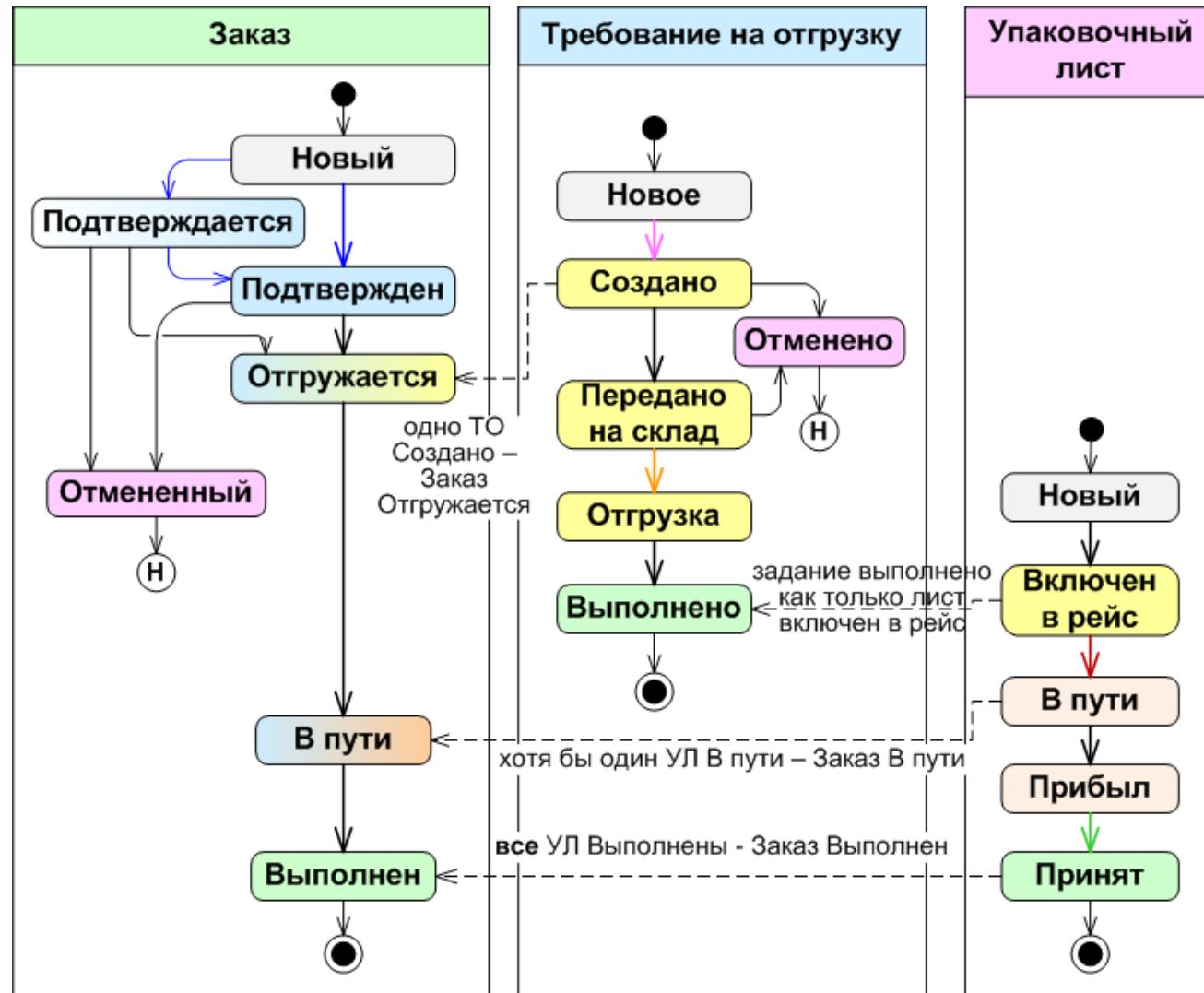
- Документу приписываем **состояние**, оно определяет этап жизненного цикла
 - Какие действия можно совершать и кому
 - Кто отвечает за обработку документа
- Возможные изменения состояний документа образуют **граф переходов** – State machine diagram



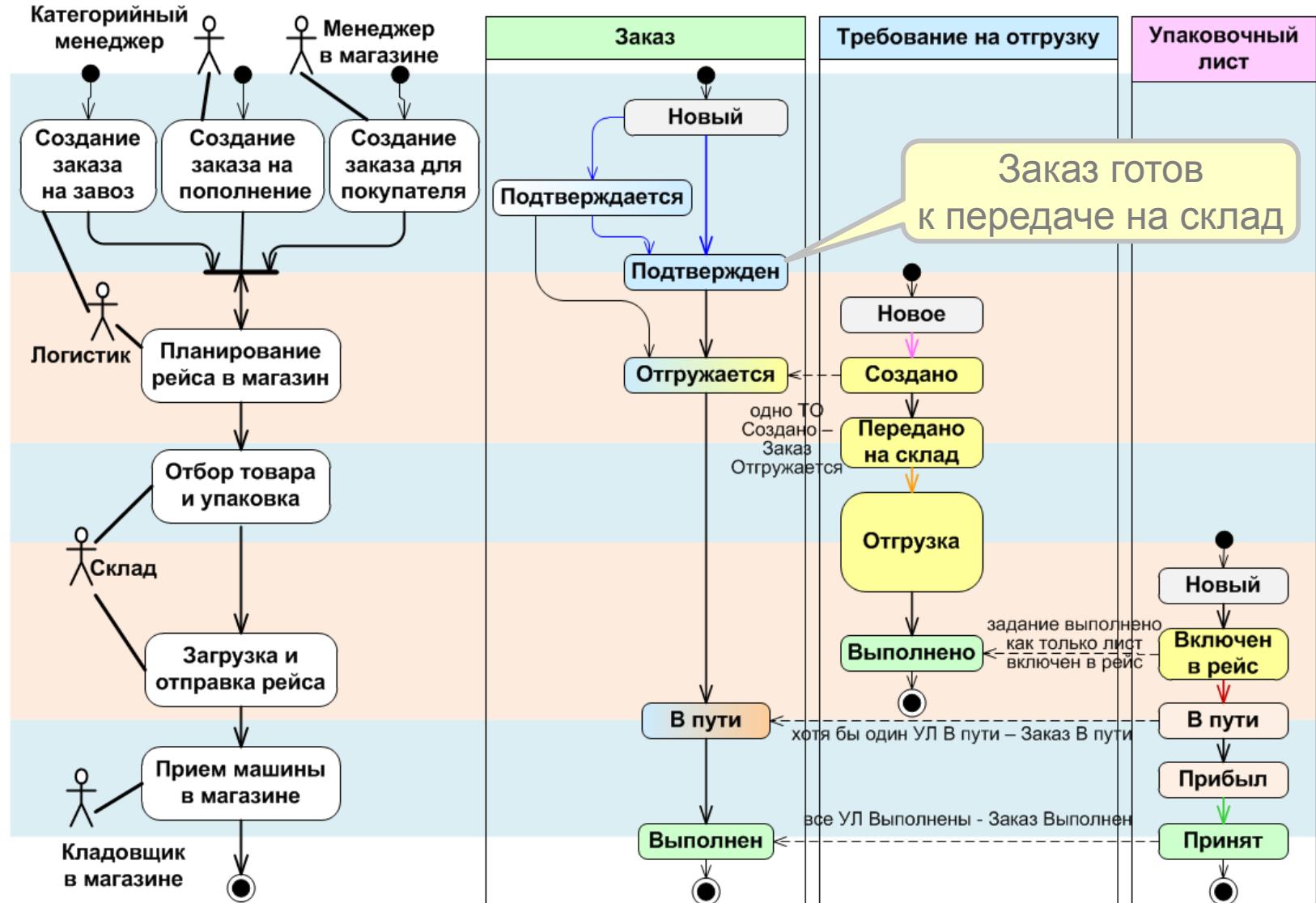
Язык ООП «с расширениями».
Названия состояний и переходов –
на языке бизнеса



Наглядно представляет сложный документооборот



Переходы документов и бизнес-функции



И в заключение...

- ➡ Нет схем правильных и неправильных – есть **облегчающие коммуникацию** и бесполезные для нее
- ➡ Нотация должна пониматься всеми, а не только автором
- ➡ Визуальное описание эффективнее текста, но не заменяет его

Вопросы? Обращайтесь!



Вакансии архитекторов и аналитиков

Пишите на hr@custis.ru, подходите с вопросами

Максим Цепков
mtsepkov.org

