

# Лекции 9-10

## **«Проектирование взаимодействий»**

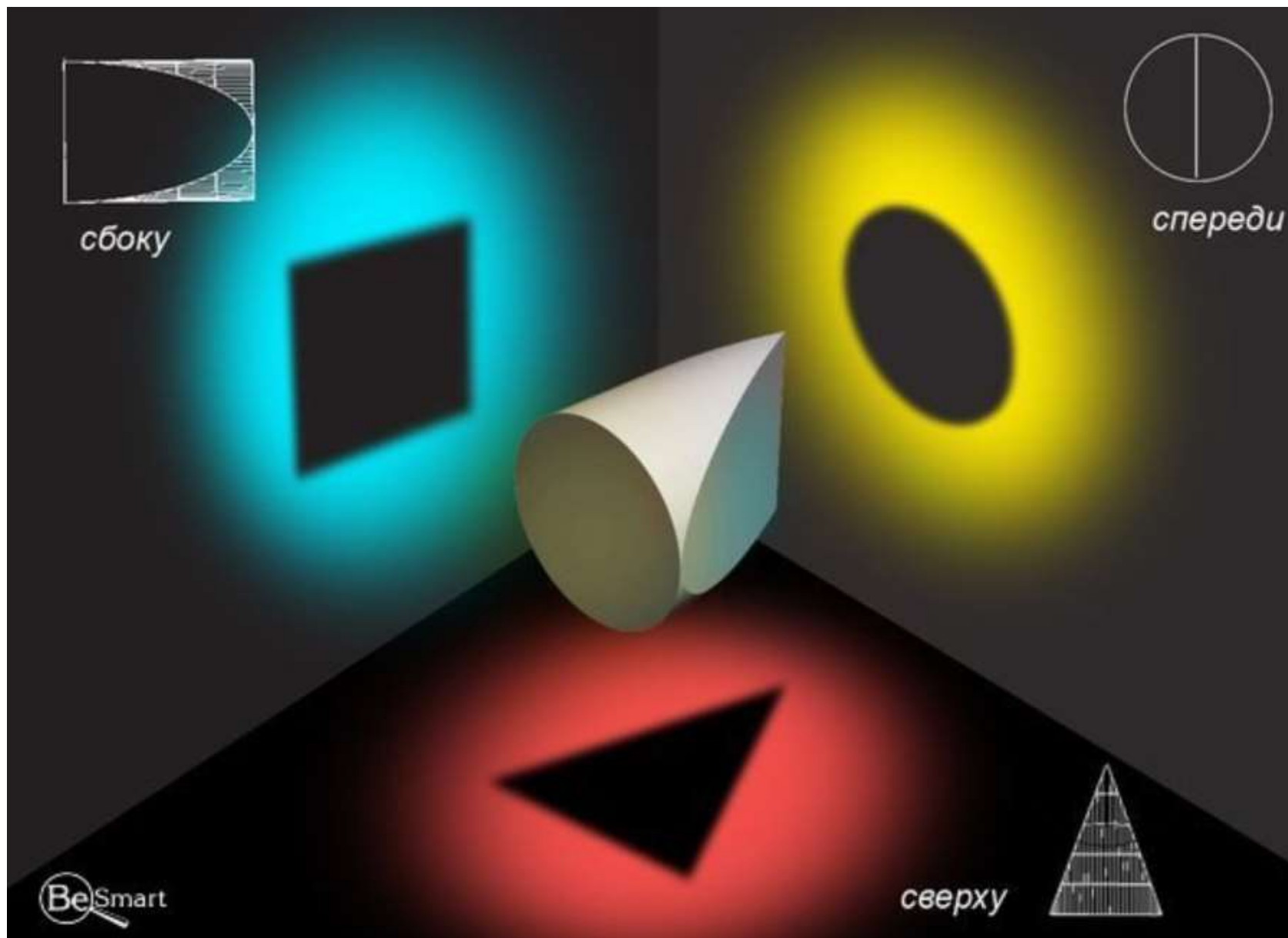
Овчинников П.Е.  
МГТУ «СТАНКИН»,  
ст.преподаватель кафедры ИС

# **Лекция 9**

## **«Вертикальное и горизонтальное взаимодействия»**

Овчинников П.Е.  
МГТУ «СТАНКИН»,  
ст.преподаватель кафедры ИС

# Точка зрения определяет все



# Взаимодействие (философия)

**Взаимодей́ствие** — базовая философская категория, отражающая **процессы** воздействия **объектов** (**субъектов**) друг на друга, их изменения, взаимную обусловленность и порождение одним объектом других

По сути, взаимодействие представляет собой разновидность **опосредованной** или **непосредственной**, **внутренней** или **внешней связи**; при этом **свойства** любых **объектов** могут быть познанными или проявить себя только во взаимодействии с другими **объектами**

Философское понятие *взаимодействия*, нередко выступая в роли интеграционного фактора, обуславливает **объединение** отдельных **элементов** **в некий новый вид** целостности, и, таким образом, имеет глубокую связь с понятием структуры

Взаимодействие — объективная и универсальная форма движения, развития, которая определяет существование и **структурную организацию** любой материальной системы

# Воздействие (управление)

**Система управления** — систематизированный (строго определённый) набор средств:

- **сбора сведений** о подконтрольном **объекте** и
  - средств **воздействия** на его поведение
- предназначенный для достижения определённых целей

Объектом системы управления могут быть как технические объекты, так и люди. Объект системы управления может состоять из других объектов, которые могут иметь постоянную структуру взаимосвязей.

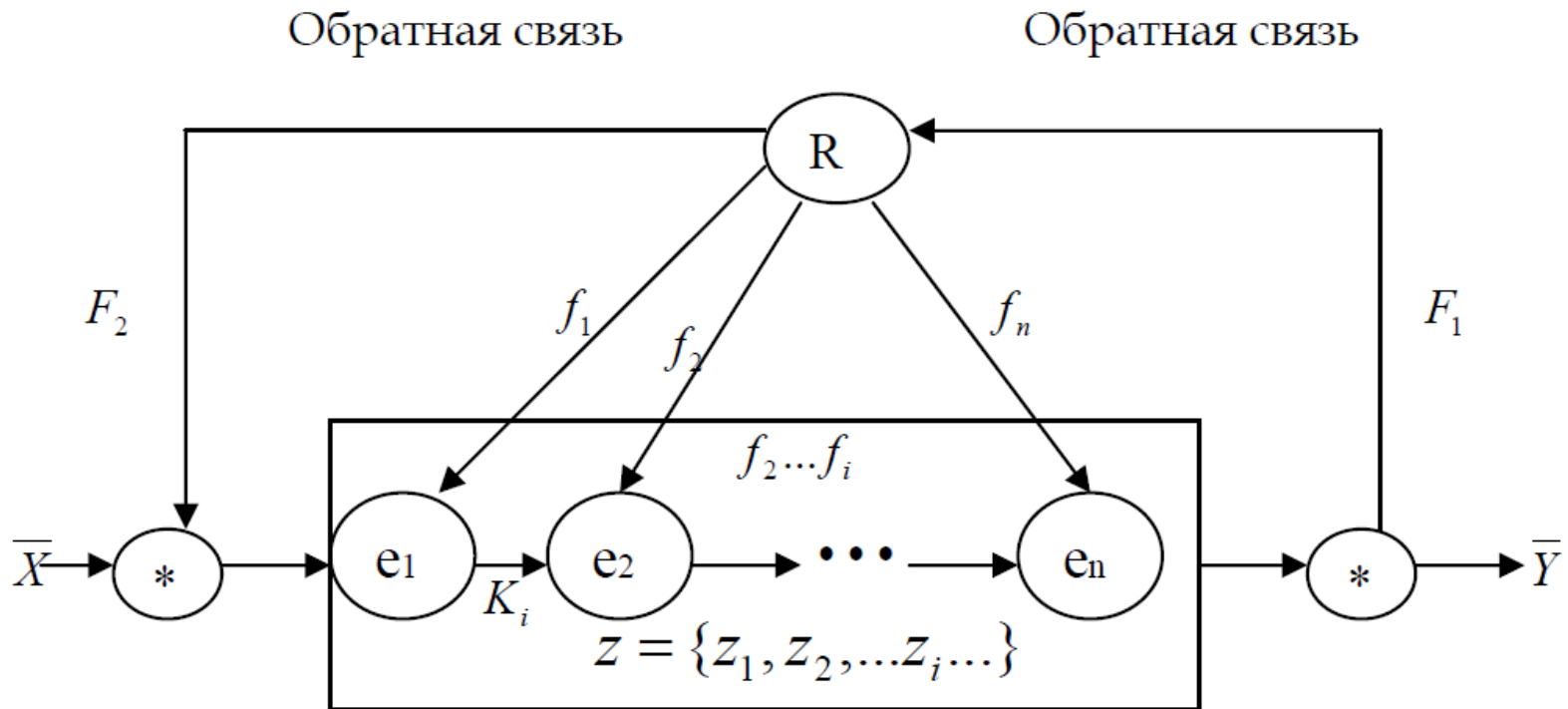
**Техническая структура управления** — устройство или набор устройств для манипулирования поведением других устройств или систем

Объектом управления может быть любая динамическая система или её модель

Состояние объекта характеризуется некоторыми количественными величинами, изменяющимися во времени, то есть переменными состояниями

# Топология в системах управления

Система  $\Sigma$  – это конечная совокупность элементов (E) и некоторого регулирующего устройства (R), которое устанавливает связи между элементами ( $e_i$ ) по преобразованию и управлению, управляет этими связями, создавая неделимую единицу функционирования. Топологически система представлена на рис. 1.



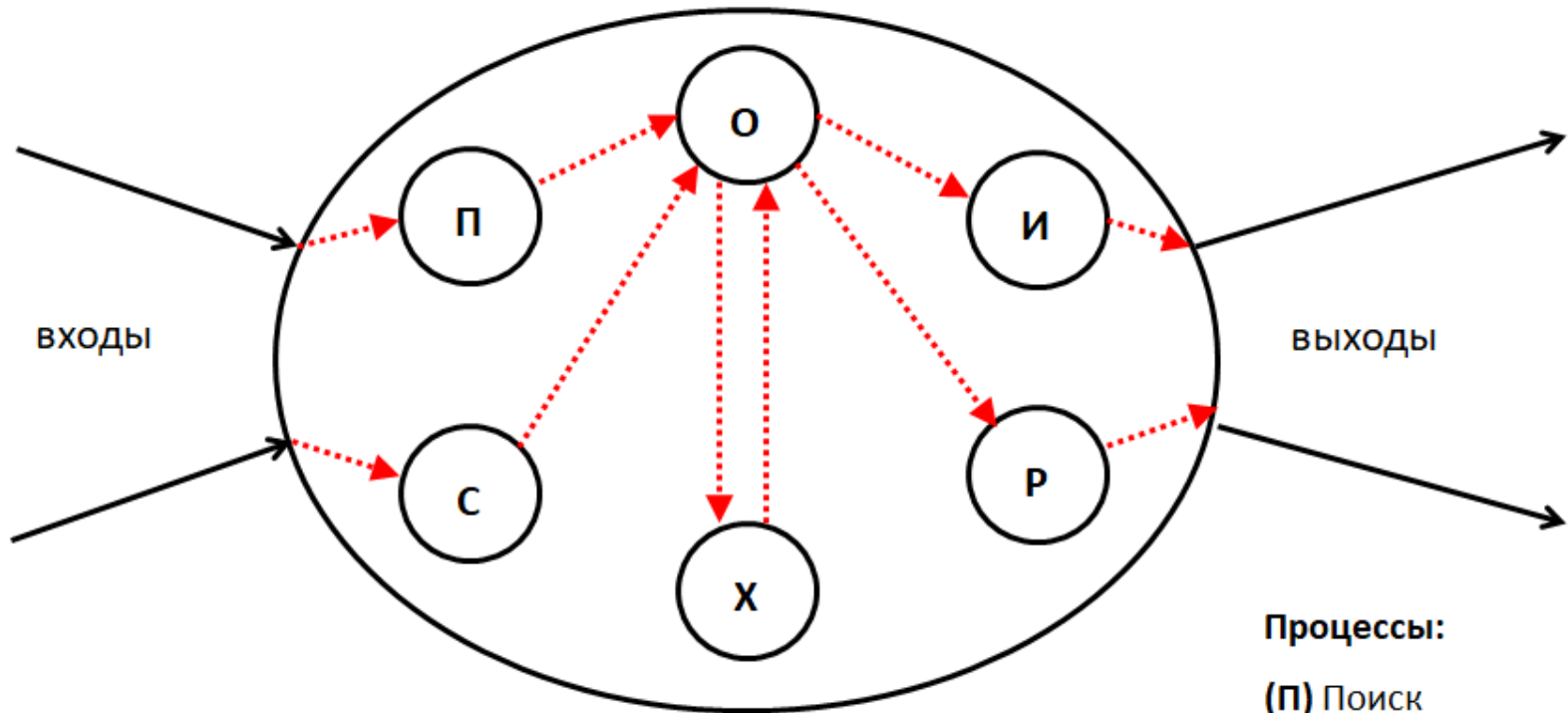
Данелян Т.Я.

Д177 ТЕОРИЯ СИСТЕМ И СИСТЕМНЫЙ АНАЛИЗ (ТСиСА): учебно-методический комплекс / Т.Я. Данелян. – М.: Изд. центр ЕАОИ, 2010. – 303 с.

ISBN 978-5-374-00324-6

# Объектно-ориентированный подход (инженерия знаний)

Внешняя среда



Процессы:

(П) Поиск

(С) Сбор

(О) Обработка

(Х) Хранение

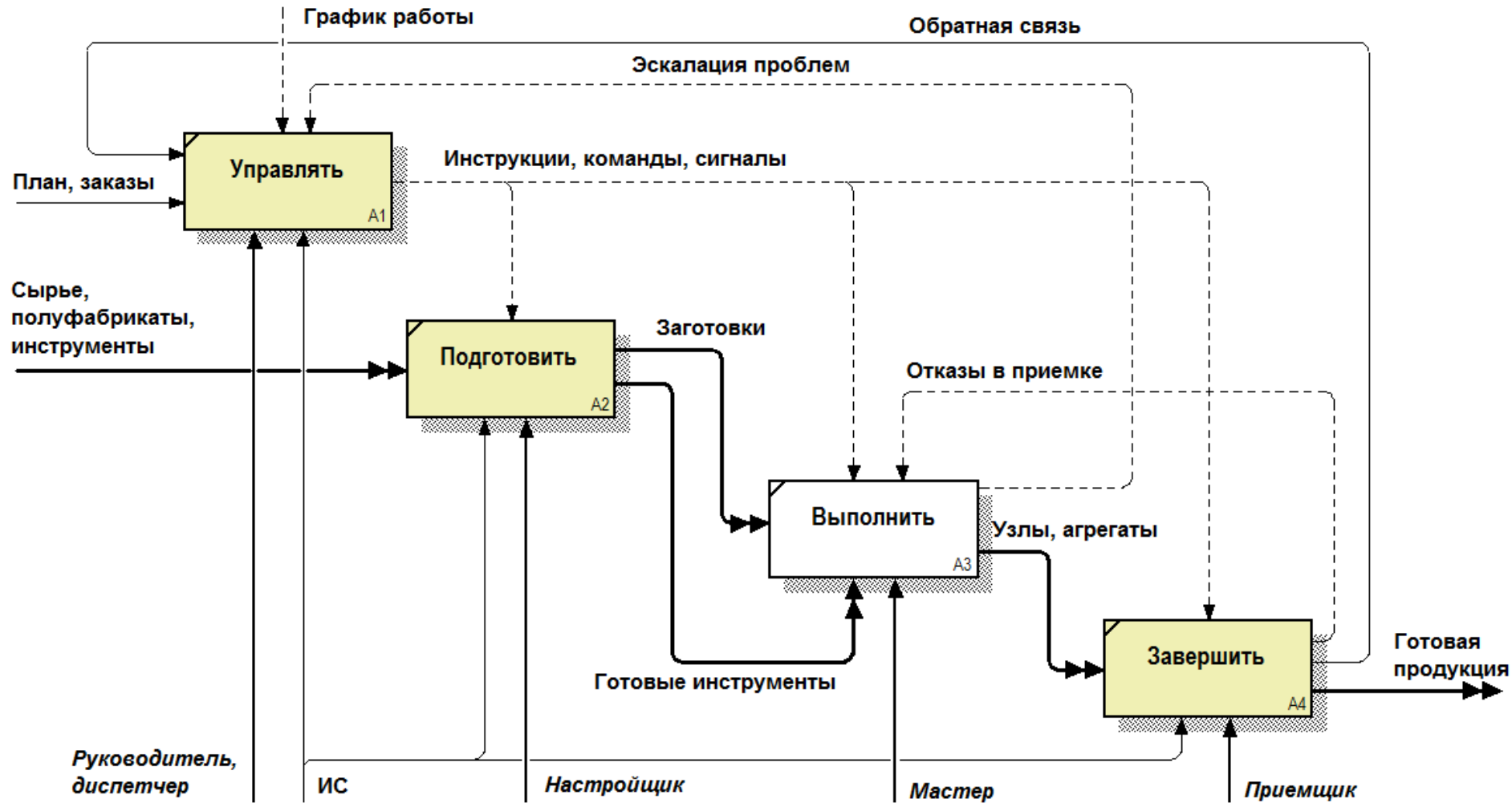
(И) Предоставление

(Р) Распространение

*Одно из решений:*

- 1. Поиск и сбор получают информацию из внешней среды*
- 2. Предоставление и распространение отправляют информацию во внешнюю среду*
- 3. Хранение взаимодействует только с обработкой*

# Прямые и обратные связи в управлении





# Вертикальное взаимодействие

Подразумевает

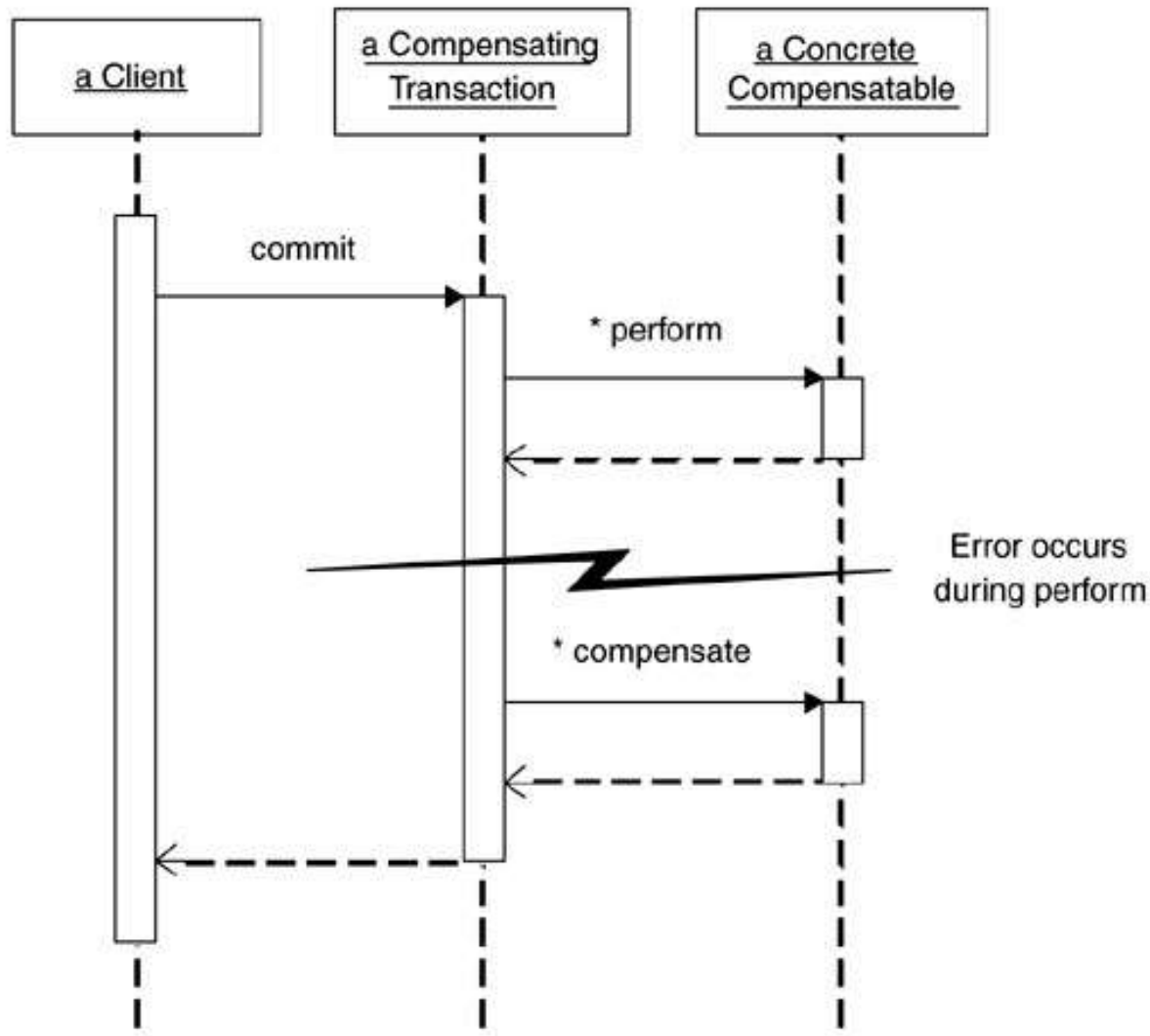
1. обмен
2. информацией

в ходе

3. совместной
4. деятельности

между

5. объектами
6. различных (!!!)
7. классов



# Горизонтальное взаимодействие

Подразумевает

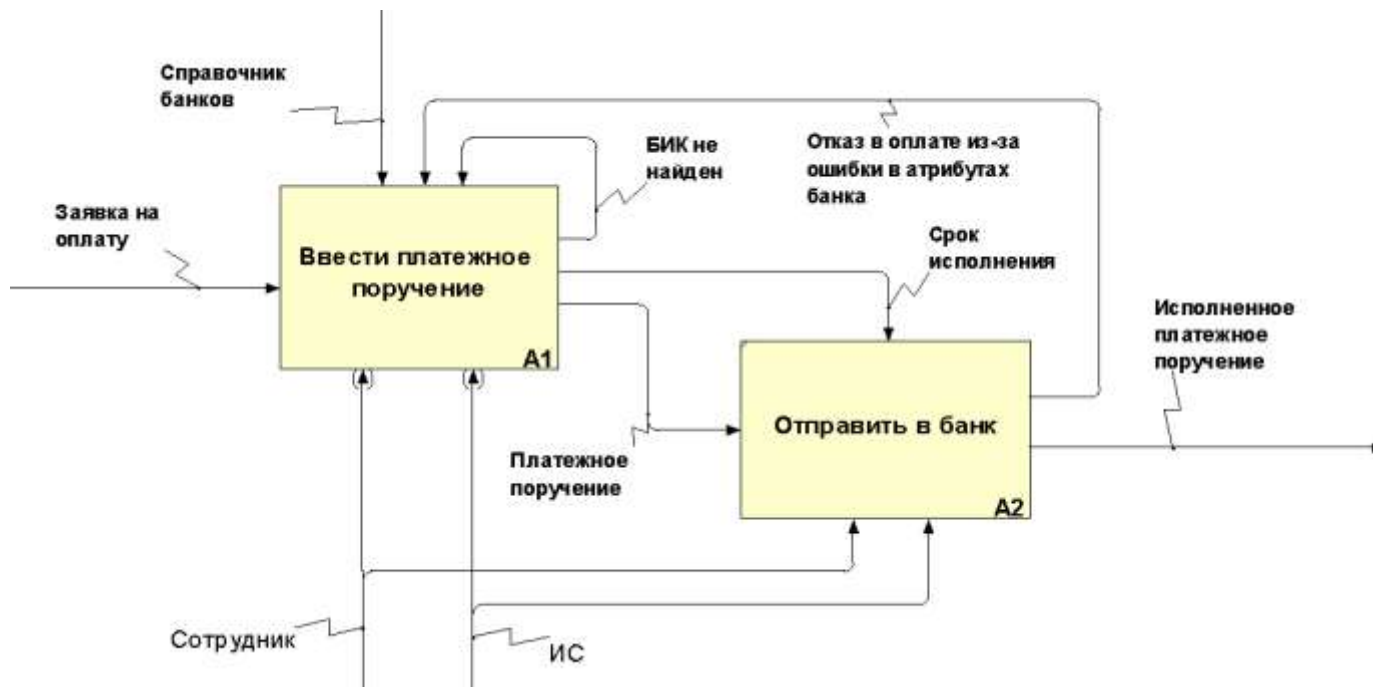
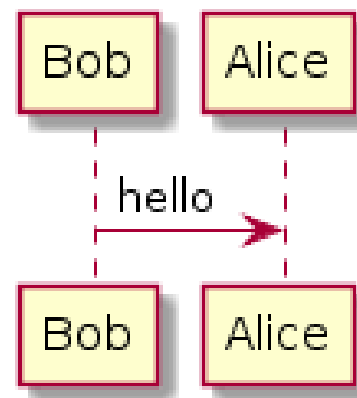
1. обмен
2. информацией

в ходе

3. совместной
4. деятельности

между

5. объектами
6. одного (!!!)
7. класса



# Взаимодействие: рабочие центры



# Взаимодействие: организационные структуры

Критерием наиболее популярной типологии организационных структур является распределение ответственности (способ группирования ответственности):

- иерархическая
- линейная
- линейно-штабная
- функциональная
- упрощённая матричная
- **сбалансированная матричная**
- **усиленная матричная**
- **проектная**
- процессная
- дивизиональная

# Взаимодействие: матричная модель

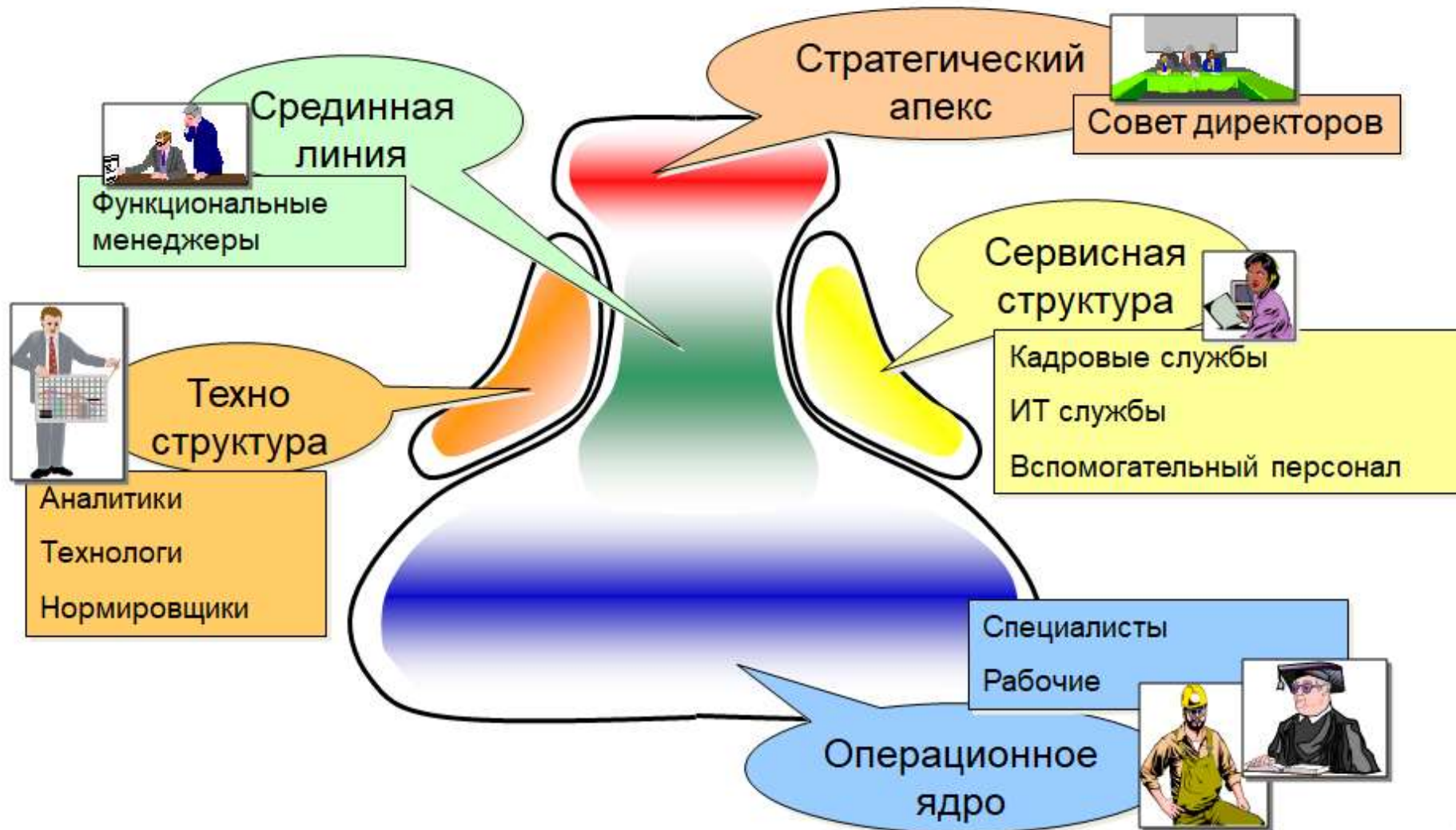


# Взаимодействие: конфигурации организаций

Предложенная [Генри Минцбергом](#) типология базируется на выделении шести основных структурных элементов организации:

- **операционное ядро** организации — осуществляет основные процессы по созданию ценности для конечного потребителя
- **стратегическая вершина** — руководство организации, осуществляющее, формирование [миссии](#), стратегических целей и стратегии деятельности организации
- **средняя линия** — промежуточное звено между руководством и операционным ядром
- **техноструктура** — объединяет аналитиков и специалистов, организующих и поддерживающих информационные потоки, формально организующих взаимодействие подразделений и контроль за их деятельностью
- **вспомогательный персонал** — службы, обеспечивающие функционирование остальных элементов организации
- **идеология** — атмосфера организации, связанная с её традициями.

# Взаимодействие: структура конфигурации



# Взаимодействие: сверхструктуры

В теории Г.Минцберга выделяется 6 типов сверхструктур (стереотипов):

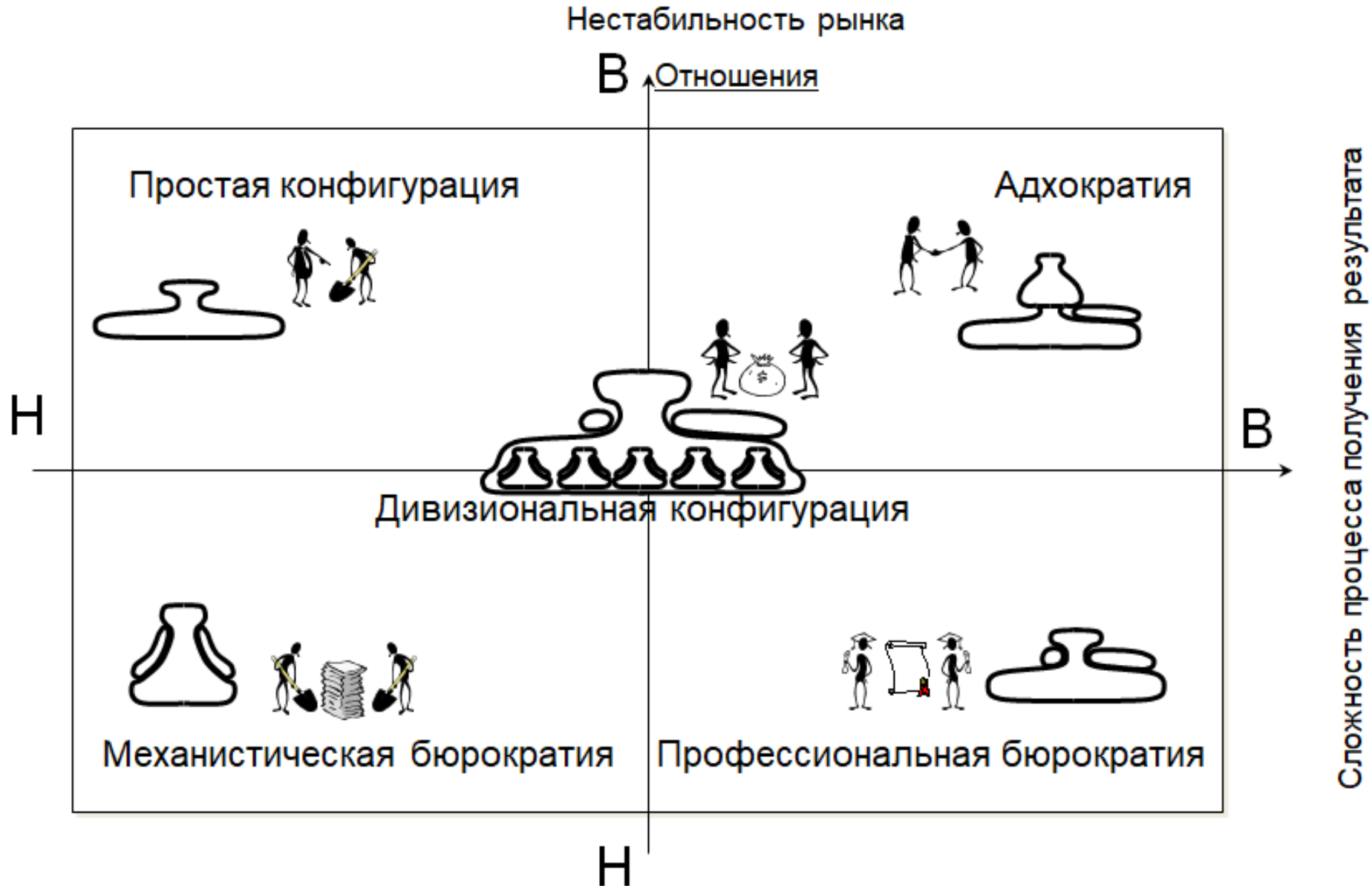
- **простая** структура — основной частью выступает **стратегическая вершина** и организация стремится к **централизации**
- **машинная бюрократия** — во главе управления стоит **техноструктура** с доминирующим стремлением к **стандартизации**
- **профессиональная бюрократия** — власть принадлежит **операционному ядру**, наиболее ценным качеством выступает **профессионализм**
- **адхократия** — основной частью является **вспомогательный персонал**, стремящийся к **сотрудничеству** с внешними организациями
- **дивизиональная** форма — главную роль играет **средняя линия** за счёт увеличения роли среднего звена
- **миссионерская** форма — **ценности и идеология** ставятся во главу управления организацией



# Взаимодействие: стереотипы



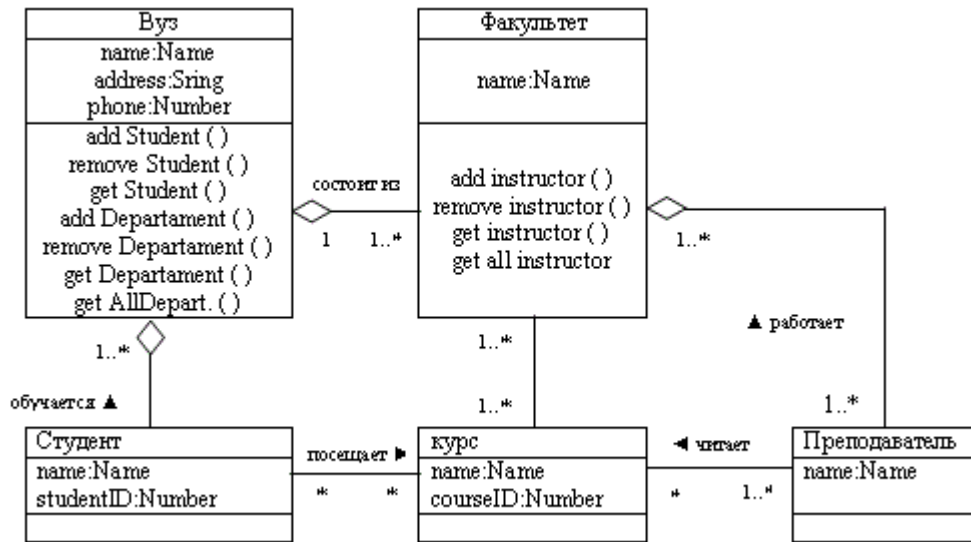
# Взаимодействие: стереотипы



# Взаимодействие: диаграммы

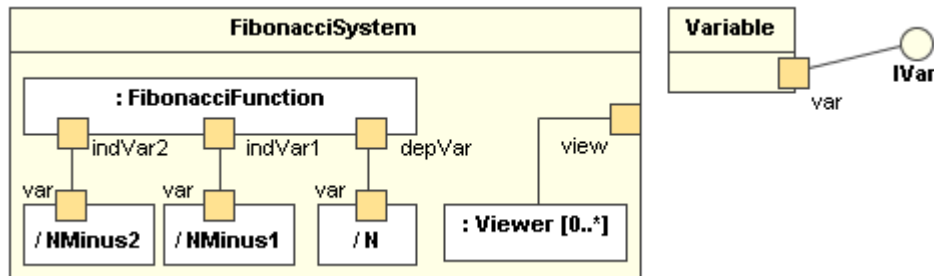
## Диаграмма классов

Классы, их атрибуты,  
операторы, взаимосвязь



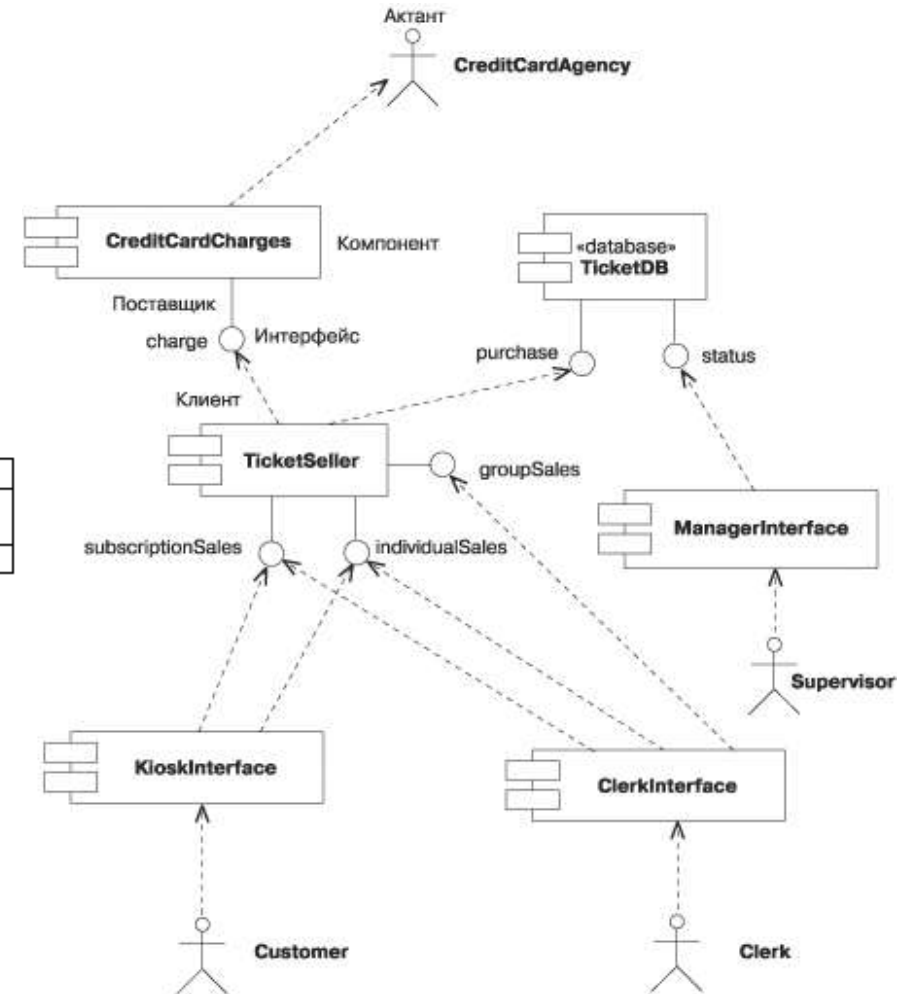
## Диаграмма составной структуры

Внутренняя структура классов



## Диаграмма компонентов

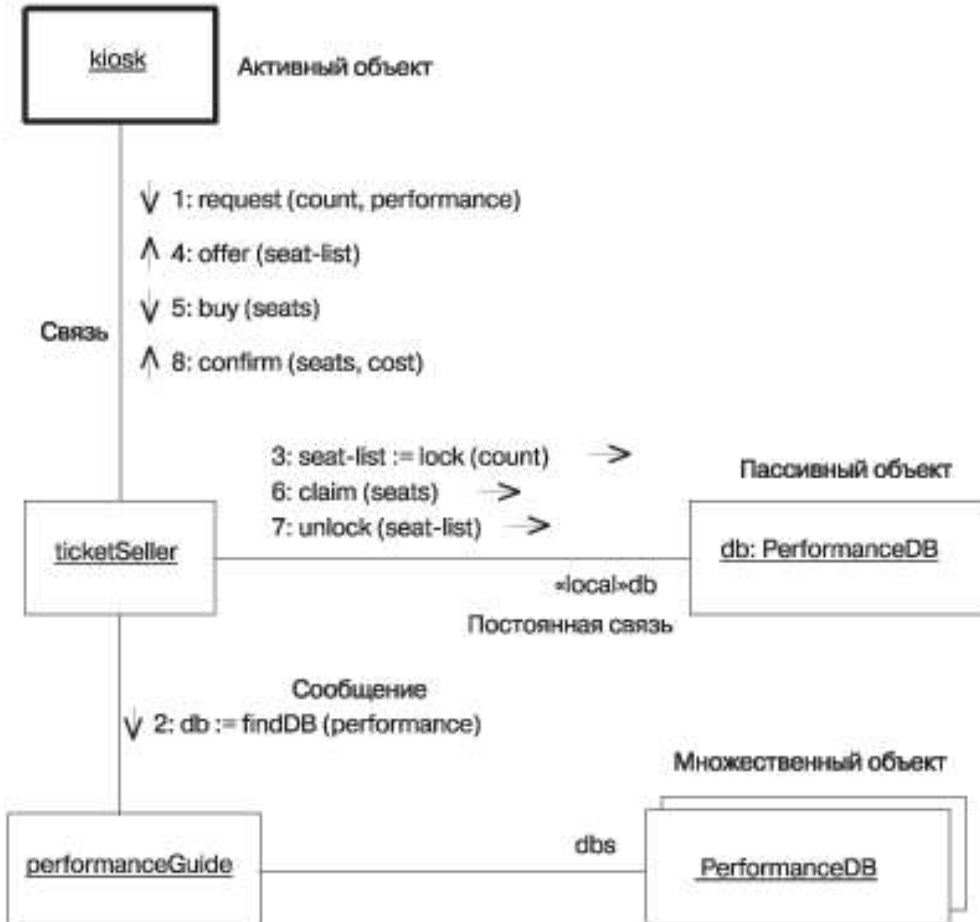
Компоненты системы, их  
взаимосвязь



# Взаимодействие: диаграммы

## Диаграмма взаимодействия

Объекты, участвующие во взаимодействии,  
их связи



## Диаграмма пакетов

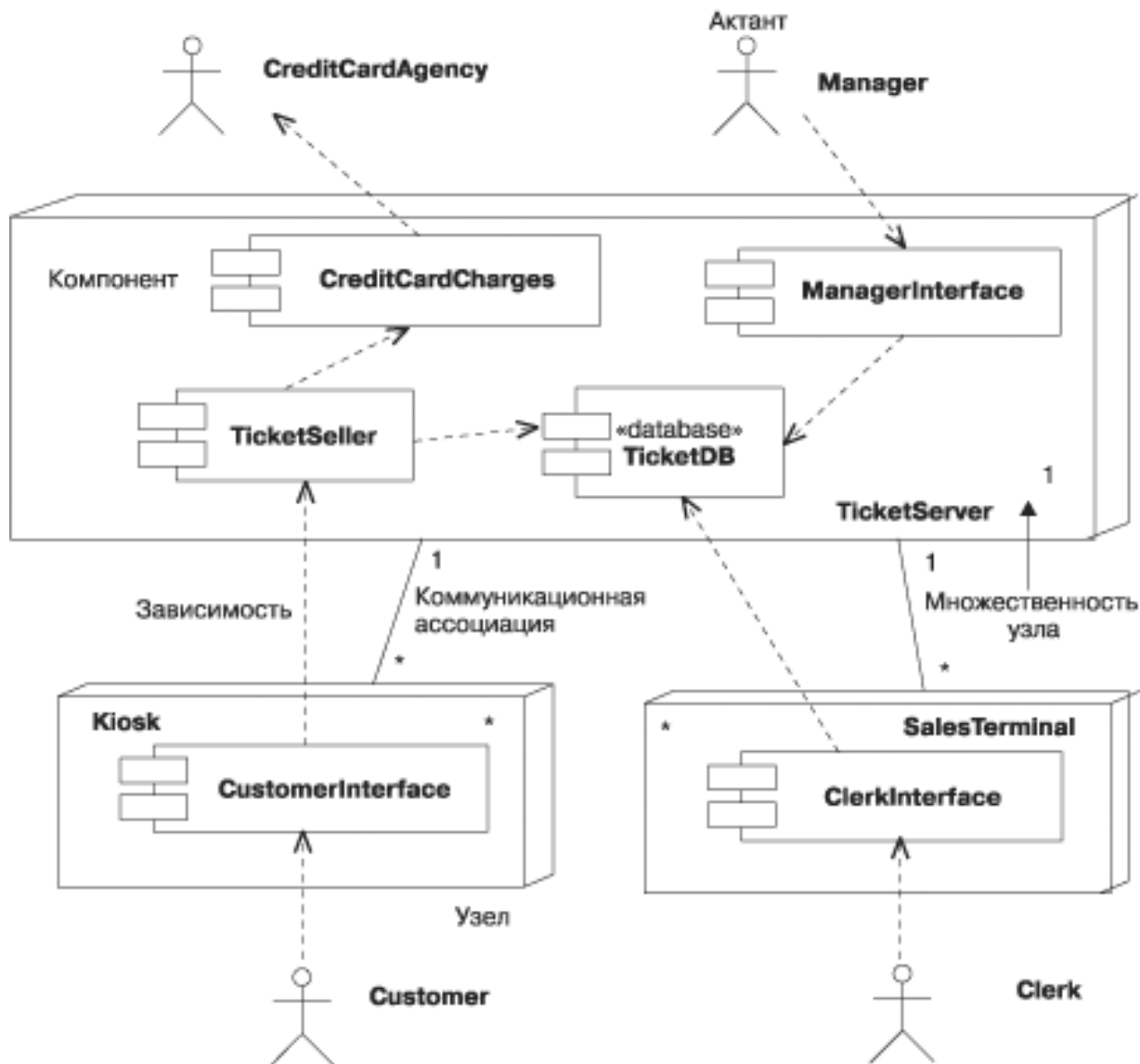
Зависимости между пакетами



# Взаимодействие: диаграммы

## Диаграмма развёртывания

*Конфигурация узлов, где производится обработка информации*



# Взаимодействие: диаграммы

Диаграмма объектов  
Экземпляры классов (объекты)  
с указанием текущих значений  
их атрибутов и связей между  
объектами.

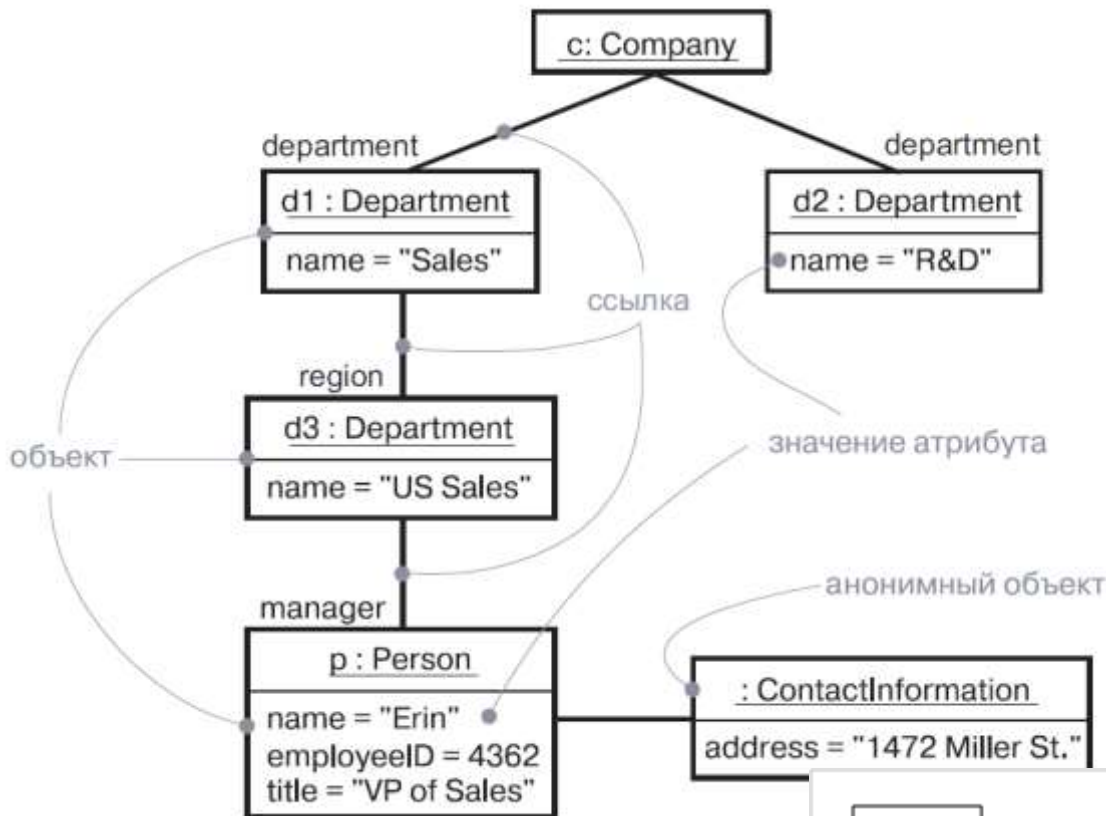
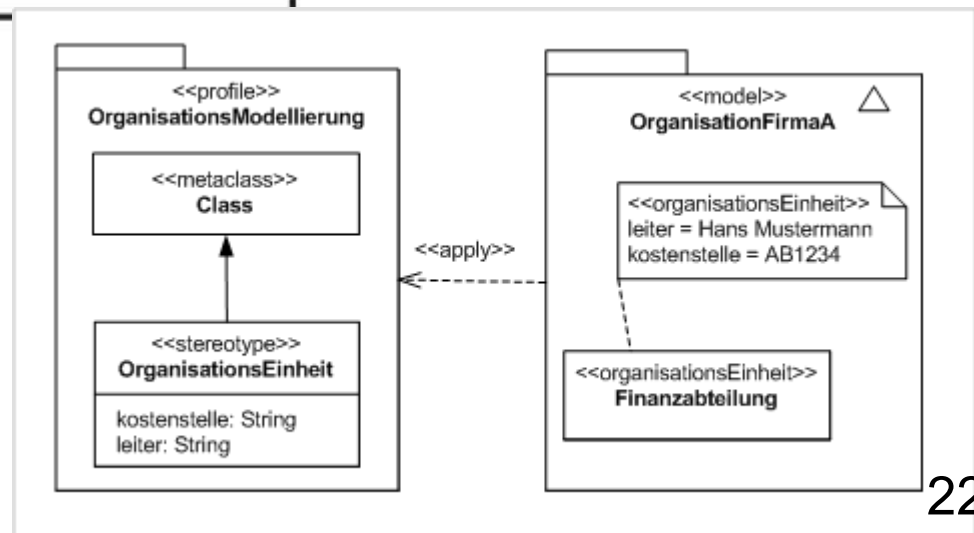


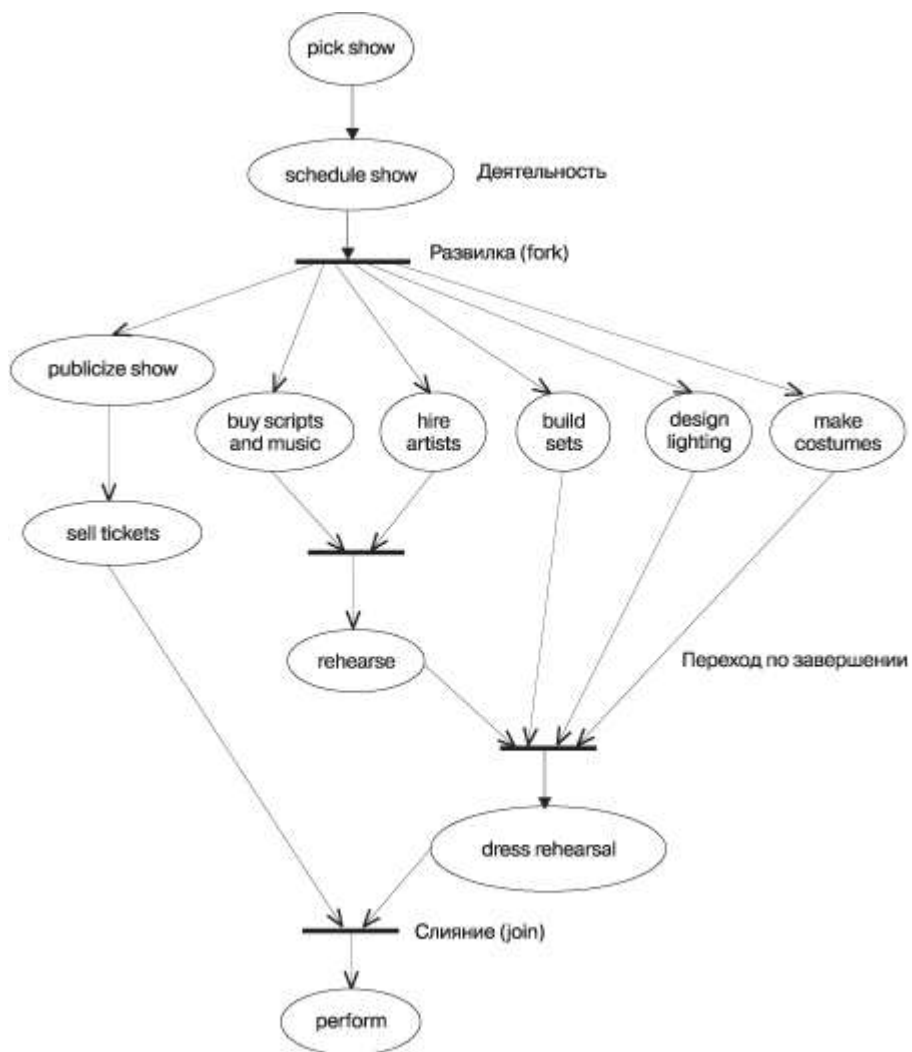
Диаграмма  
профилей



# Взаимодействие: диаграммы

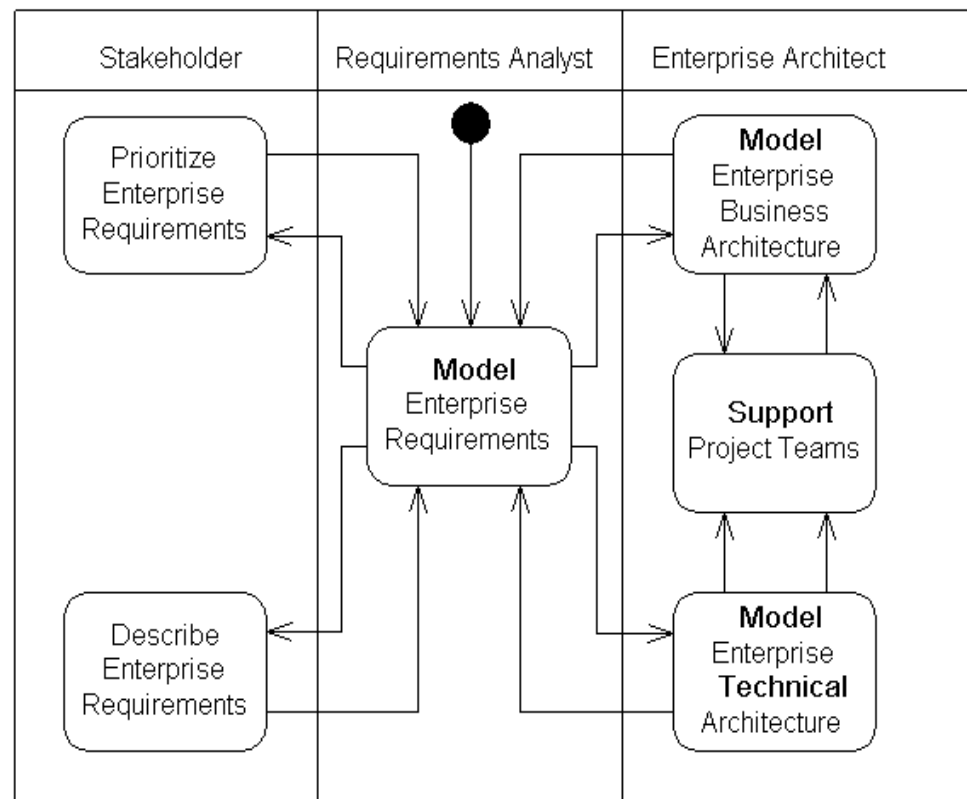
## Диаграмма деятельности

*Разложение некоторой деятельности на её составные части*



## Диаграмма деятельности

(«плавательные дорожки»)



# Взаимодействие: диаграммы

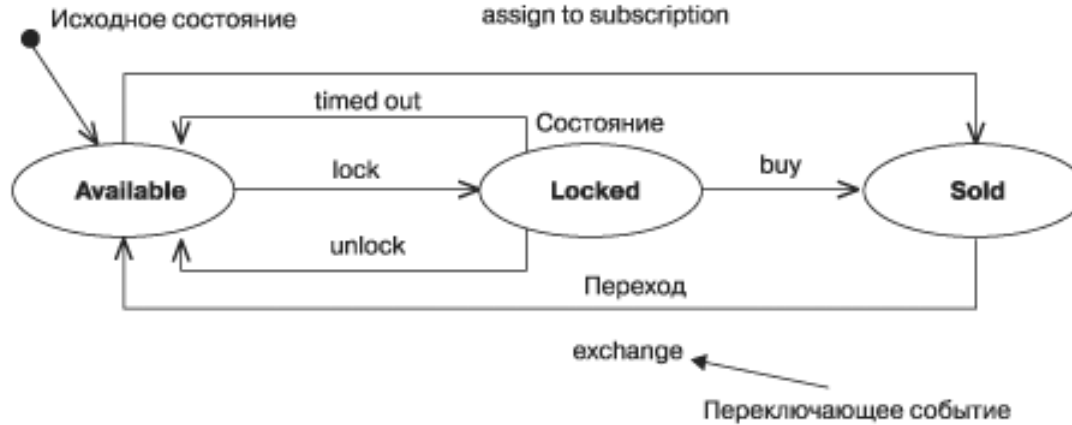
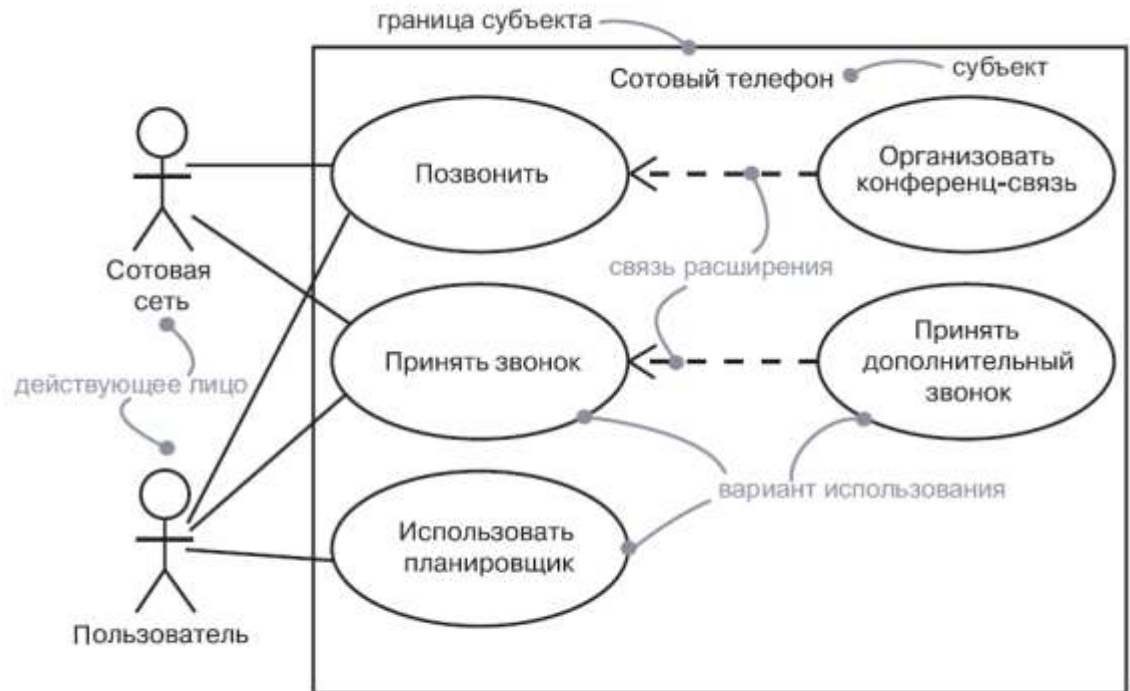


Диаграмма состояний  
*Все возможные состояния системы под воздействием различных действий или событий*

## Диаграмма прецедентов

*Отношения между действующими лицами и прецедентами*

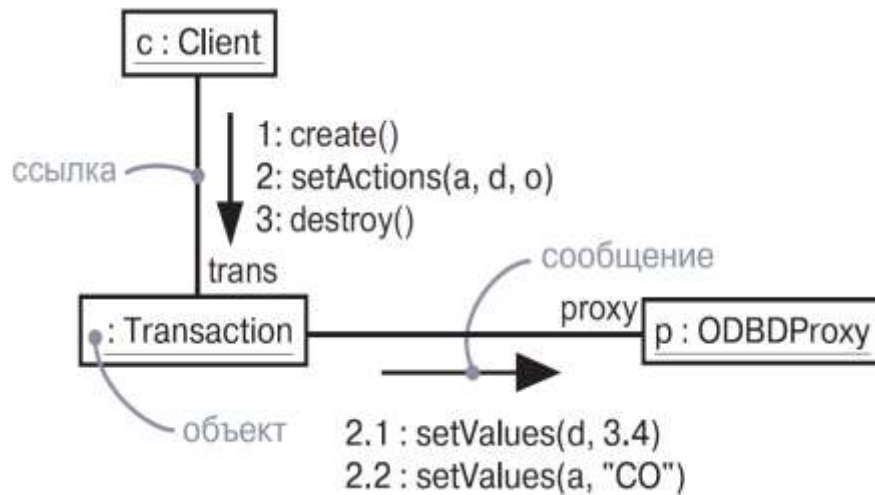




# Взаимодействие: диаграммы

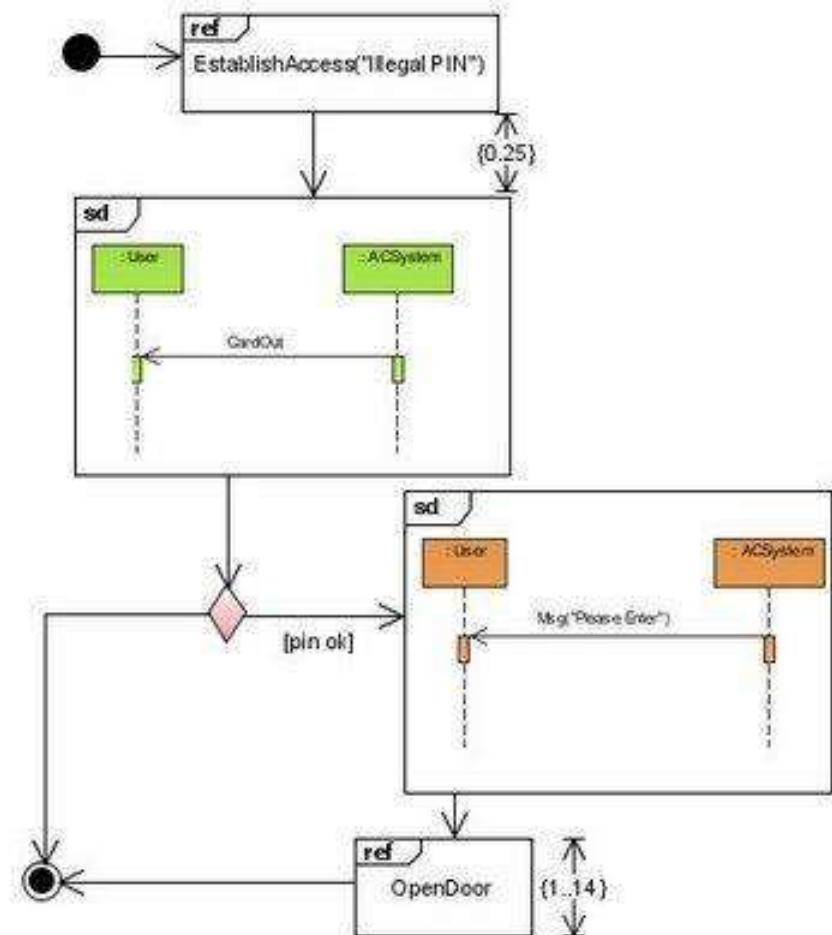
## Диаграмма коммуникации

*Взаимодействия между частями составной структуры или ролями кооперации*



## Диаграмма обзора взаимодействия

Диаграмма деятельности + диаграмма последовательности



# Диаграмма синхронизации

# Диаграмма последовательности

```

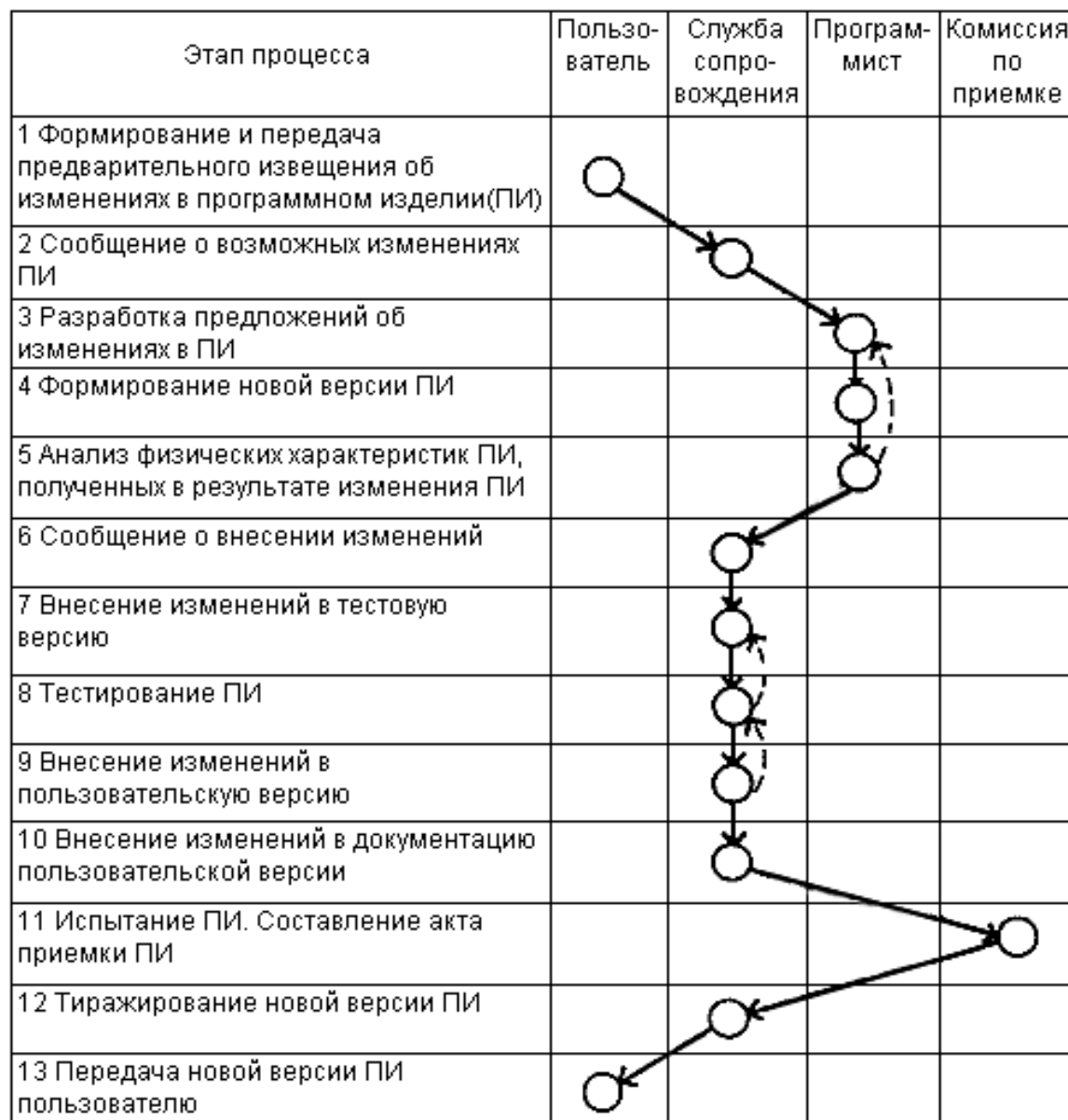
sequenceDiagram
    participant Client as c:Client
    participant Transaction as :Transaction
    participant Proxy as p:ODBCProxy

    Client->>Transaction: create()
    activate Transaction
    Transaction->>Transaction: setActions(a, d, o)
    activate Transaction
    Transaction->>Proxy: setValues(d, 3.4)
    deactivate Transaction
    Transaction->>Proxy: setValues(a, "CO")
    deactivate Transaction
    Proxy-->>Client: committed
    deactivate Proxy
    Client->>Transaction: destroy()
    deactivate Transaction
    destroy Transaction
  
```



# Взаимодействие: диаграммы

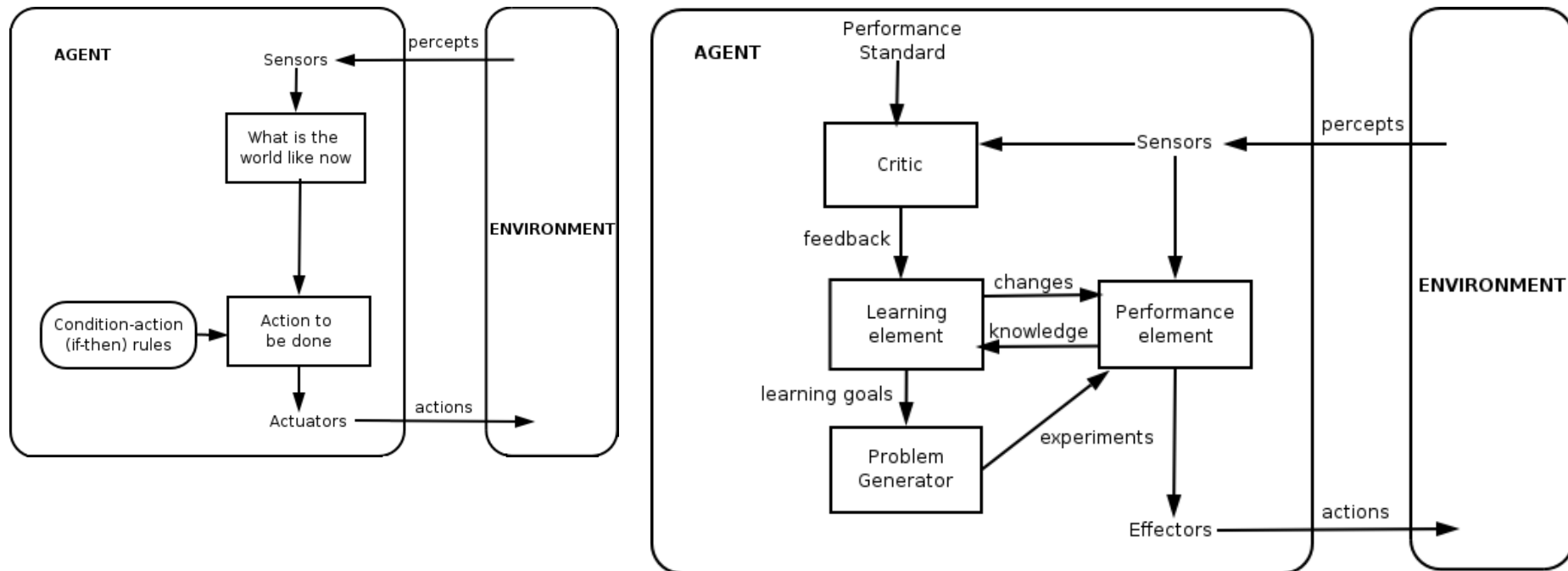
Планирование и контроль хода выполнения технологических процессов переработки данных



# Взаимодействия: мультиагентные системы

**Многоагентная система (МАС, англ. Multi-agent system)** — это система, образованная несколькими взаимодействующими интеллектуальными агентами

Многоагентные системы могут быть использованы для решения таких проблем, которые сложно или невозможно решить с помощью одного агента или монолитной системы.



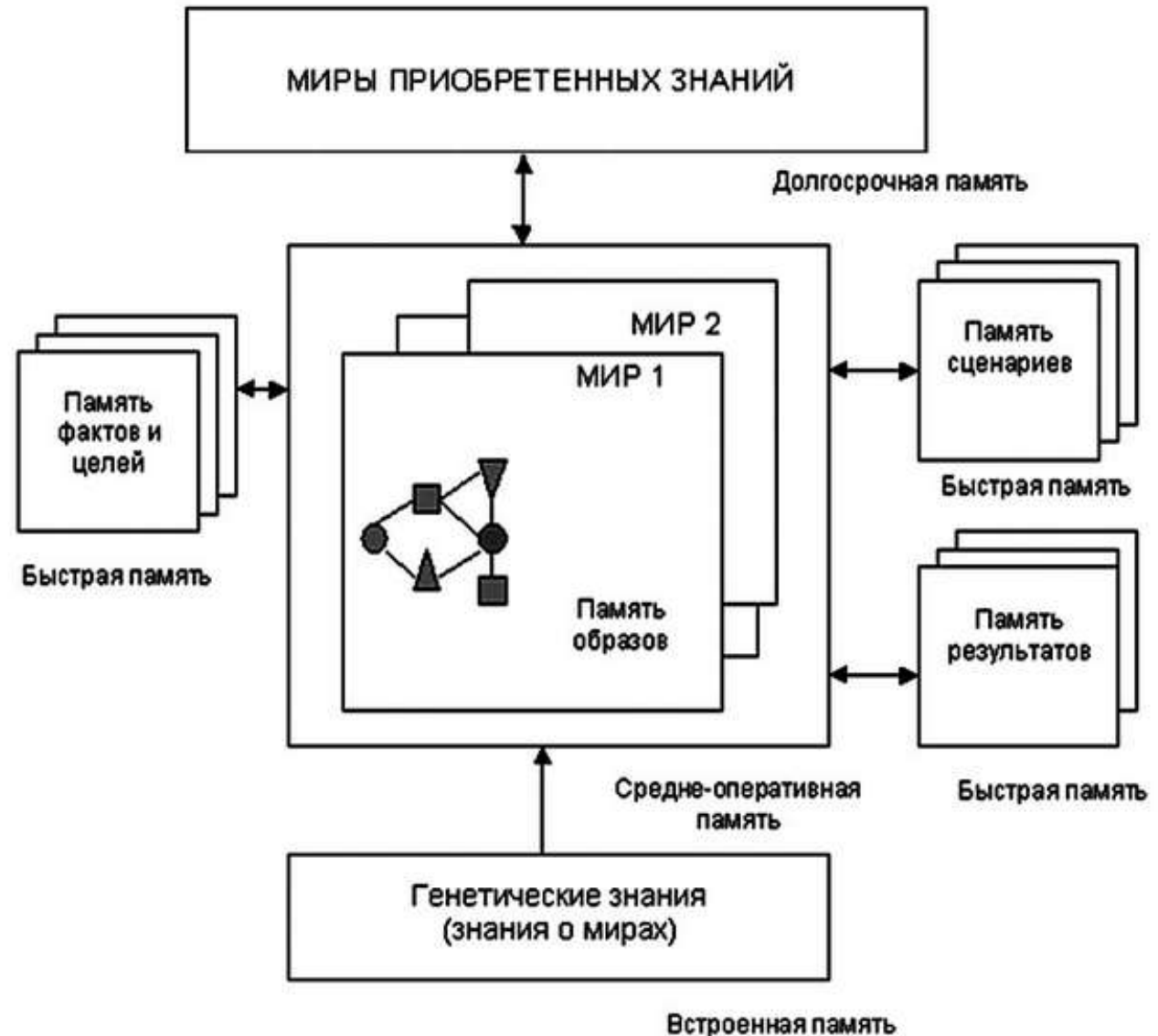
MAC

<http://www.intuit.ru/studies/courses/13833/1230/lecture/24081>

# Взаимодействия: мультиагентные системы

Концепция "Агентов и Миров" реализует формирование **общего мира деятельности** кооперирующих сторон и **миров деятельности** каждой из них, путем создания единой комплексной среды

В этом подходе **мир действий** — это **модель среды деятельности**, базирующаяся на знаниях



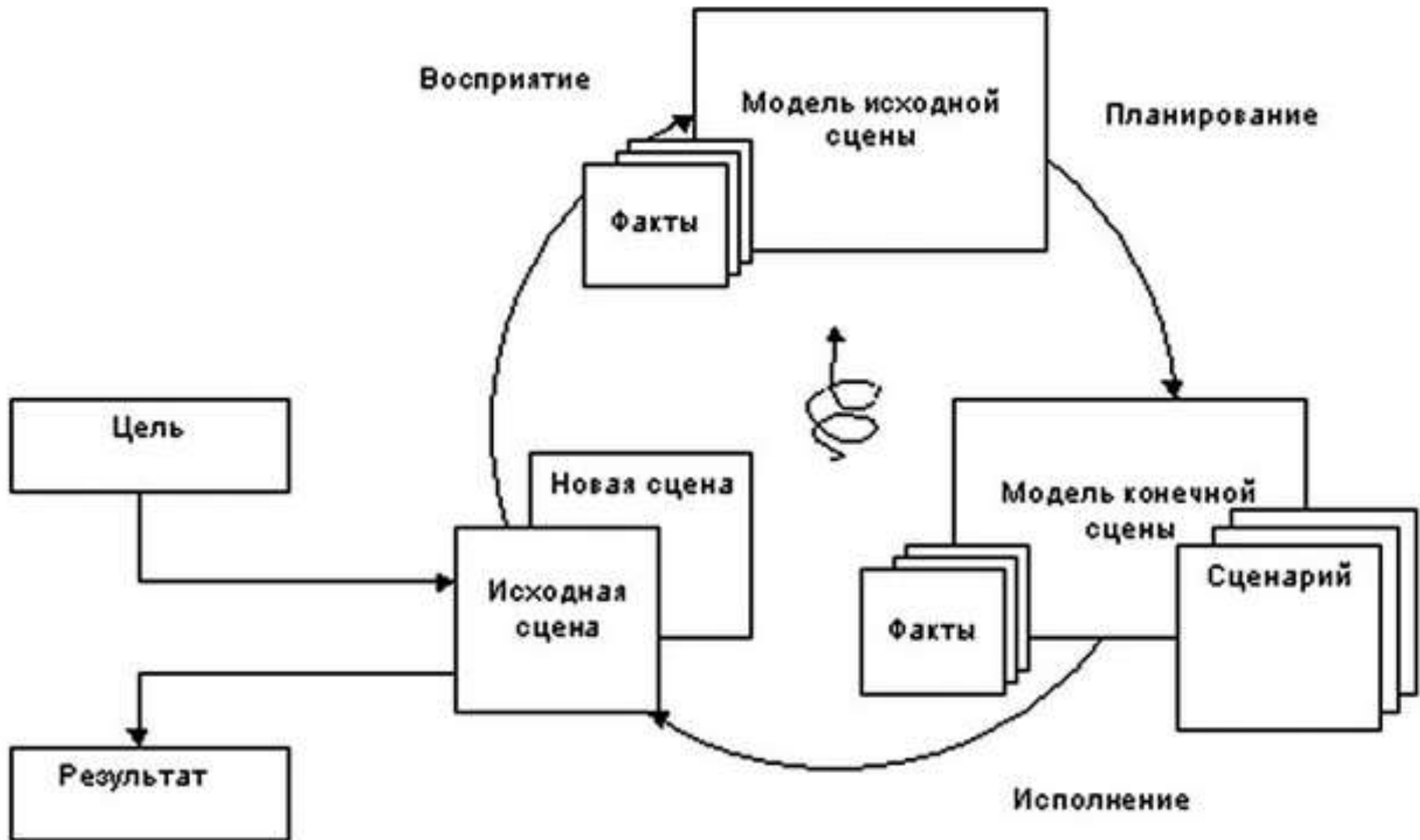
# Взаимодействия: мультиагентные системы

Важным элементом при создании мультиагентных систем является **язык коммуникации** агентов — Agent Communication Language, который определяет **типы сообщений**, которыми могут обмениваться агенты

В рамках парадигмы коммуникации между агентами, кооперация между ними достигается за счет ACL, **языка контента** и **онтологии**, которые определяют набор **базовых концепций**, используемых в сообщениях кооперации

Онтология здесь выступает синонимом понятия API (Application Programming Interface), т.е. она определяет конкретный **интерфейс** интеллектуальных агентов

# Взаимодействия: мультиагентные системы



# Взаимодействие: инициатор

Федеральный закон от 27.07.2006 N 149-ФЗ (ред. от 21.07.2014) «Об информации, информационных технологиях и о защите информации»

**Информация** - сведения (сообщения, данные) независимо от формы их представления

## Статья 10. Распространение информации или предоставление информации

1. В Российской Федерации **распространение** информации осуществляется свободно при соблюдении требований, установленных законодательством Российской Федерации.

3. При использовании для **распространения** информации средств, позволяющих определять получателей информации, в том числе почтовых отправлений и электронных сообщений, лицо, распространяющее информацию, обязано обеспечить получателю информации возможность отказа от такой информации.

4. **Предоставление** информации осуществляется в порядке, который устанавливается соглашением лиц, участвующих в обмене информацией.



# Взаимодействие: инициатор

**Вытягивающее** производство (англ. *pull production*) — схема организации производства, при которой **объёмы** продукции и **сроки** её изготовления на каждом производственном этапе определяются исключительно **потребностями последующих этапов** (в конечном итоге — потребностями заказчика)

Отпуск материалов в производство со складов выполняется **по требованию потребителя**, к моменту использования материала в производственных операциях

Решения по **пополнению запасов** материалов на складах принимаются на самих складах, а не центральной службой или заводом

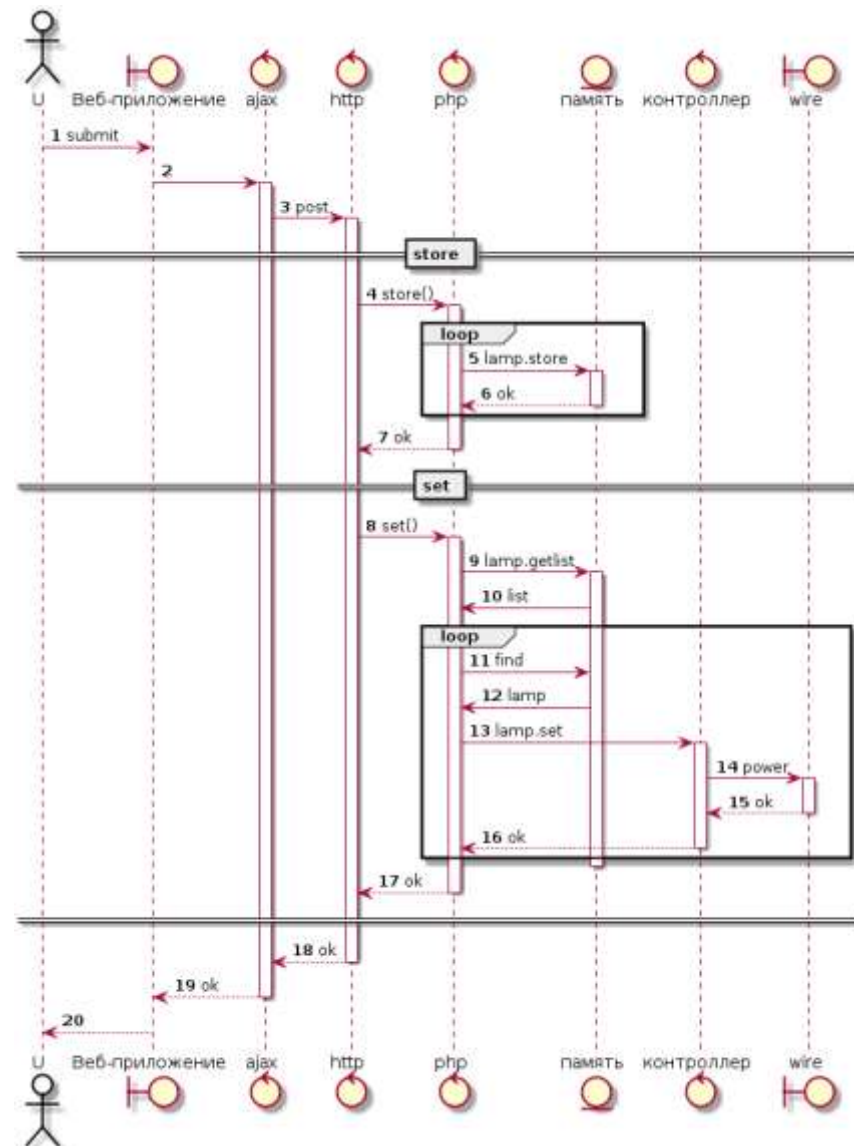
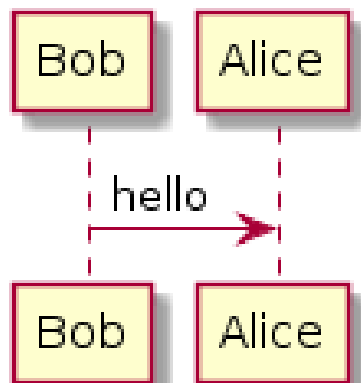
Наряду со схемой «точно вовремя» вытягивающая производственная схема — часть концепции бережливого производства

# Взаимодействие: инициатор

## Лабораторная 3 - определение способов существования информации

- **Описание хранилищ данных**
  - Определение хранилищ данных в DFD (таблицы, файлы, массивы)
  - Открытие проекта в веб-браузере, получение и сохранение модели в развернутом виде
  - Загрузка модели в репозиторий личного проекта
  - Добавление ссылки на модель в файл readme.md
- **Описание взаимодействия участников автоматизируемой деятельности**
  - Формирование списка участников (ролей, модулей, хранилищ)
  - Определение **инициаторов взаимодействия**
  - Создание диаграммы последовательности
  - Сохранение текста и рисунка диаграммы последовательности
  - Добавление ссылок на тексты и рисунки в файл readme.md
- **Завершение идентификации всех потоков**
  - Построение ERD (диаграммы классов без атрибутов) для всех потоков
  - Загрузка модели и диаграммы классов в репозиторий
  - Завершение редактирование readme.md (Отчет о 3 лабораторных)

# Взаимодействие: инициатор



# Планирование запасов: принятие риска

## Риск (Risk)

вероятное для проекта **событие**, наступление которого может как **отрицательно**, так и **положительно** отразиться на результатах проекта.

## Требование (Requirement)

определенные условия или характеристики, которым должны соответствовать или которые должны иметь система, продукт, услуга, результат или элемент

## Допущение (Assumption)

**фактор**, который **считается верным** для проекта без привлечения доказательств

## Ограничение (Constraint)

**сдерживающий фактор**, влияющий на ход исполнения проекта

## Дефект (Defect)

Несовершенство или упущение в элементе проекта, из-за которого этот элемент не соответствует требованиям или характеристикам и должен быть **либо исправлен, либо заменен**

## Проблема (Issue)

**Обсуждаемая** или еще **не решенная** проблема, или проблема, по которой существуют **противоположные мнения** и **разногласия**

# Планирование запасов: принятие риска

В управлении **неблагоприятными рисками (threats)** применяются следующие основные **стратегии** (методы):

Уклонение (Avoidance)

$$P(X) = 0$$

Снижение (Mitigation)

$$P(X) < P(T)$$

Передача (Transference)

$$P(X) = P(X1)*P(X2) < P(T)$$

Принятие (Acceptance)

$$P(X) = P(T)$$

В управлении **благоприятными рисками (opportunity)** применяются следующие основные стратегии (методы):

Использование (Exploit)

$$P(X) = 1$$

Усиление (Enhance)

$$P(X) > P(O)$$

Совместное использование (Share)

$$P(X) = P(X1)*P(X2) > P(O)$$

Игнорирование (Ignore)

$$P(X) = P(O)$$

$P(T)$  = **исходная** вероятность **угрозы**

$P(O)$  = **исходная** вероятность **возможности**

$P(X)$  = **целевая** вероятность риска **после применения** стратегии

Лайфхак (противоположные события):

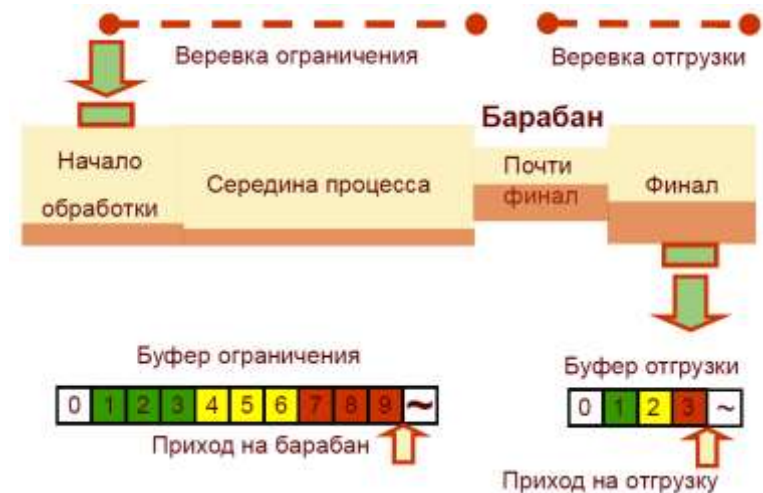
Для любой угрозы **T** существует такая возможность **O**, что

$$P(O) + P(T) = 1$$

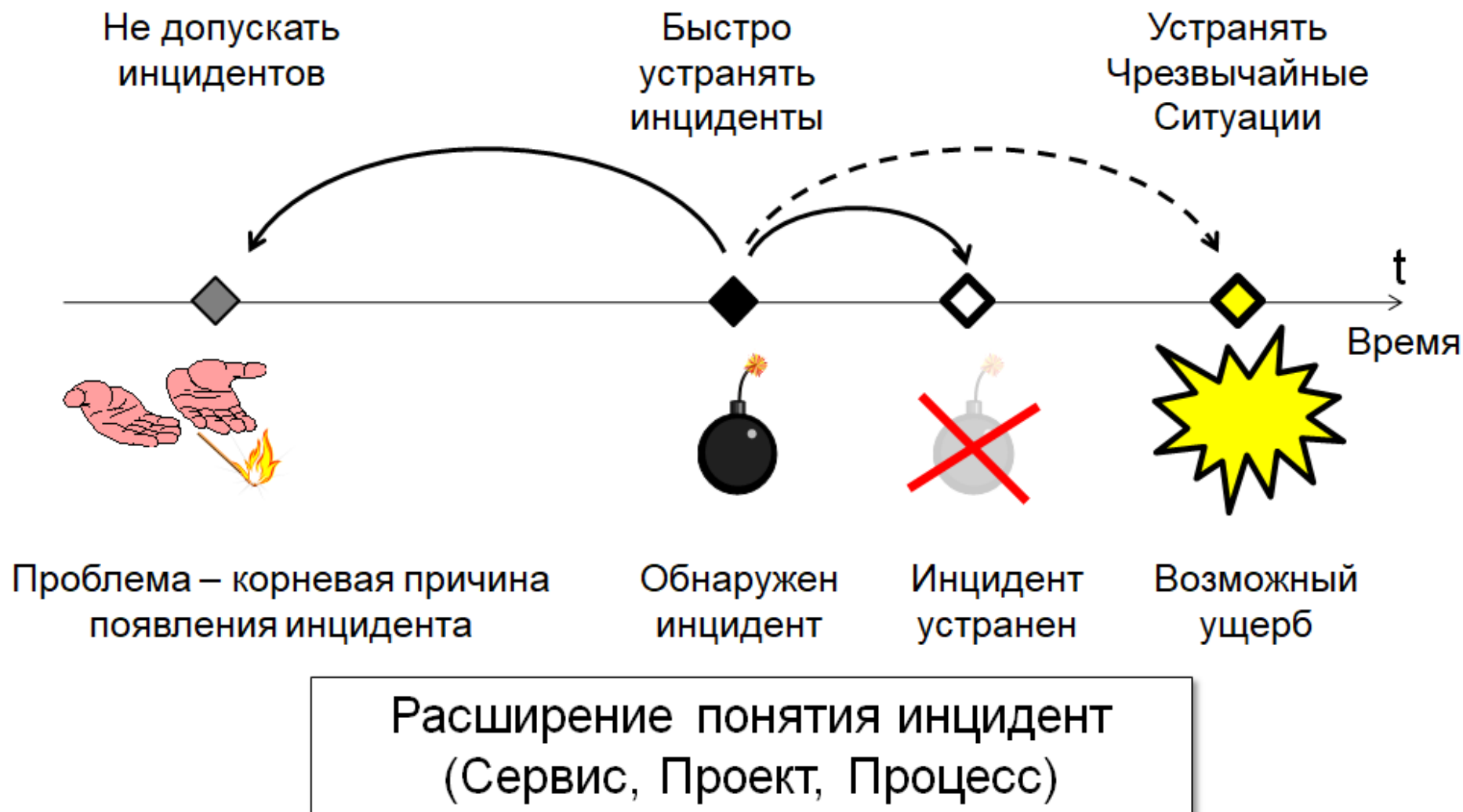
# Планирование запасов: принятие риска

В **управлении рисками** используются следующие основные виды **работ (действий)**:

- Заблаговременное воздействие на **вероятность возникновения** риска
- Заблаговременное воздействие на **степень влияния** риска на
  - **содержание** проекта (Scope)
  - **бюджет** проекта (Cost)
  - **сроки** проекта (Schedule) или на
  - **качество результата** проекта (Quality)
- **Планирование реакции** на неблагоприятное событие (Contingency Plan)
- **Обход** неожиданно возникших неблагоприятных событий (Workaround)



# Планирование рисков: стереотипы



# Планирование рисков: стереотипы

## Errors Will Happen!

When executing JavaScript code, different errors can occur.

Errors can be coding errors made by the programmer, errors due to wrong input, and other unforeseeable things.

### Example

In this example we have written alert as adddler to deliberately produce an error:

```
<p id="demo"></p>

<script>
try {
  adddler("Welcome guest!");
}
catch(err) {
  document.getElementById("demo").innerHTML = err.message;
}
</script>
```

Try it Yourself »

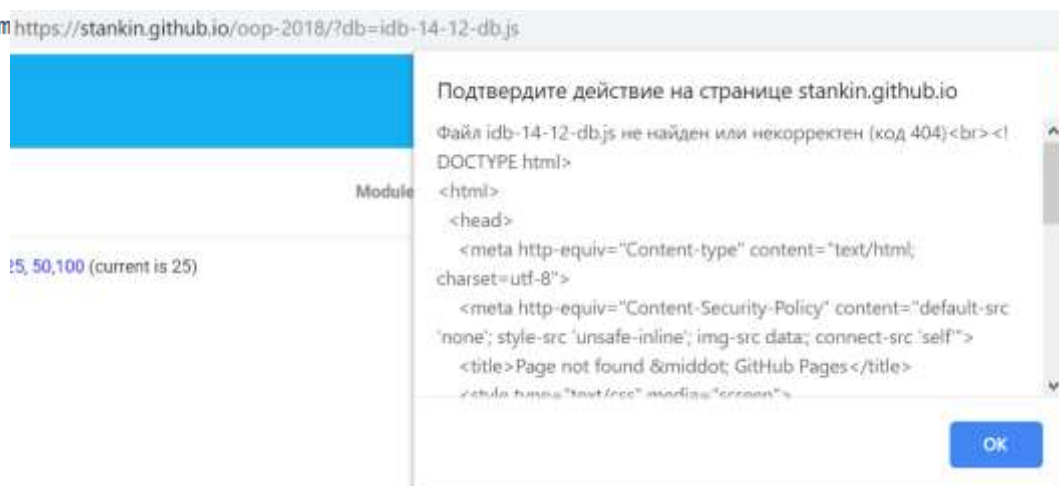


# Планирование рисков: стереотипы

```
jQuery("#save").click(function () {  
    if (jQuery('#form').jVal()) {  
        jQuery.ajax({  
            type: "POST",  
            url: "saveuser.do",  
            dataType: "html",  
            data: "userId=" + encodeURIComponent(trim(document.forms[0].userId.value)),  
            success: function (response) {  
                jQuery("#usergrid").trigger("reloadGrid");  
                clear();  
                alert("Details saved successfully!!!");  
            },  
            error: function (xhr, ajaxOptions, thrownError) {  
                alert(xhr.status);  
                alert(thrownError);  
            }  
        });  
    }  
});
```

# Планирование рисков: стереотипы

```
36 var database_path = parseQueryString().db
37 if (!database_path) alert("Не указан путь к бд")
38 $http.get(database_path)
39 .then(function(response){
40     $scope.content = $scope.$$childHead.orderBy(response.data.members, 'name', false);
41     console.log($scope.$$childHead)
42     $scope.$$childHead.order('name', true);
43     $scope.title = response.data.title
44 })
45 .catch(function(error){
46     alert("Файл "+database_path+" не найден или некорректен (код "+error.status+")<br>"+error.data)
47 })
48
49 $scope.custom = {name: 'bold', description: 'grey', last_modified: 'grey'};
50 $scope.sortable = ['name', 'm1', 'mhttps://stankin.github.io/oop-2018/?db=idb-14-12-db.js'];
51 $scope.count = 25;
52 }]);
53
```



# Лекция 10

## **«Управление сроками»**

Овчинников П.Е.  
МГТУ «СТАНКИН»,  
ст.преподаватель кафедры ИС

# Управление сроками: сетевое планирование

**Сетевой анализ** (сетевое планирование) — метод анализа сроков:

- ранних и
- поздних,
- начала и
- окончания

нереализованных частей проекта, позволяет увязать выполнение различных работ и процессов во времени, получив **прогноз общей продолжительности** реализации всего проекта.

## Детерминированные сетевые методы

Диаграмма Ганта с дополнительным временным люфтом 10-20 %

Метод критического пути (МКП)

## Вероятностные сетевые методы

Неальтернативные

Метод статистических испытаний (метод Монте-Карло)

Метод оценки и пересмотра планов (ПЕРТ, PERT)

Альтернативные

Метод графической оценки и анализа (GERT)

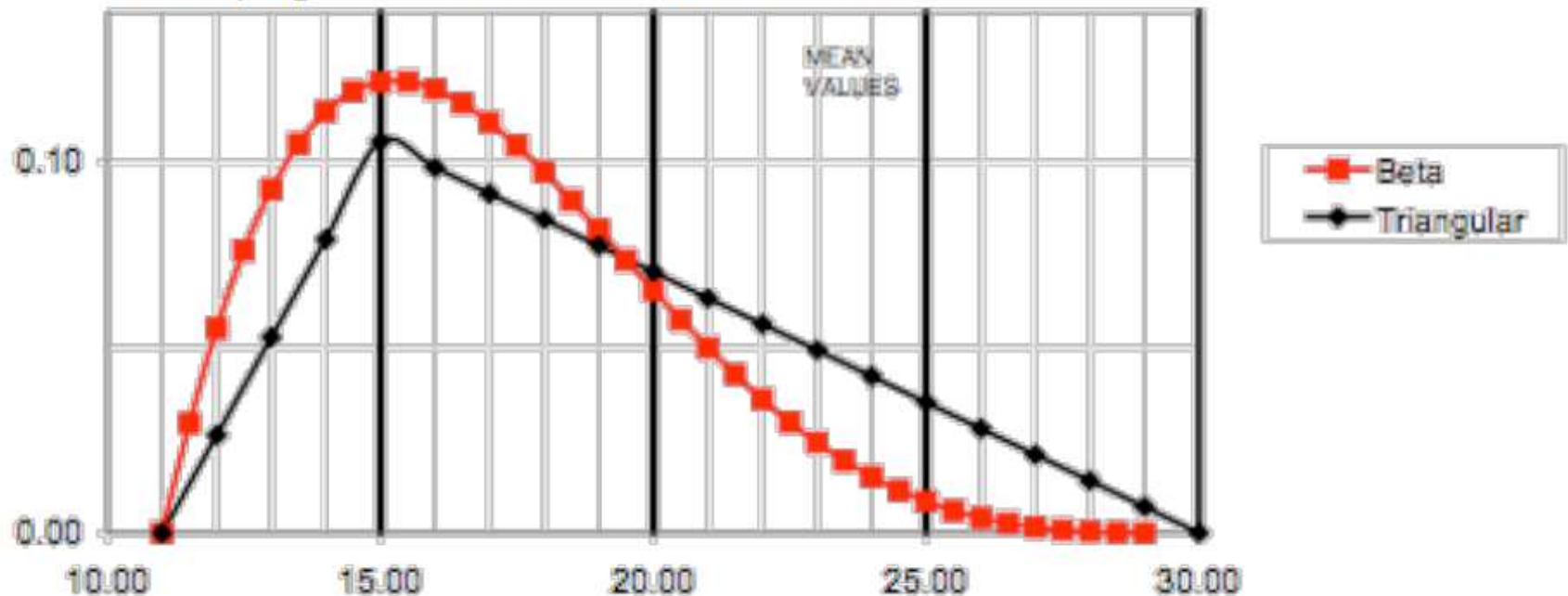
**РОЛИК**

## Метод PERT Estimation

- Обработывает три экспертных оценки срока.
  - L - «раньше не справлюсь точно, даже если повезет»;
  - H - «успею гарантированно, даже если все риски сыграют»;
  - M – «наиболее вероятно успею»
- Формулы PERT:  
$$\text{PERT Estimation } (\mu) = (L + 4M + H) / 6$$
$$\text{PERT Deviation } (\sigma) = (H - L) / 6$$
- Задача уложится в срок  $\mu + \sigma$  с вероятностью 72 %.

## Основа PERT Estimation

- Длительность задачи - случайная величина, имеющая бета-распределение.
- PERT Estimation и Deviation – матожидание и среднеквадратичное отклонение
- Между крайними оценками – 6 сигм



# Управление сроками: сетевое планирование

## Определение [\[ править | править код \]](#)

---

Пусть [распределение случайной величины](#)  $X$  задаётся [плотностью вероятности](#)  $f_X$ , имеющей вид:

$$f_X(x) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1},$$

где

- $\alpha, \beta > 0$  произвольные фиксированные параметры, и

- $B(\alpha, \beta) = \int_0^1 x^{\alpha-1} (1-x)^{\beta-1} dx$  — [бета-функция](#).

Тогда случайная величина  $X$  имеет бета-распределение. Пишут:  $X \sim B(\alpha, \beta)$ .



# Управление сроками: сетевое планирование

## Свойства задач с независимыми прогнозами

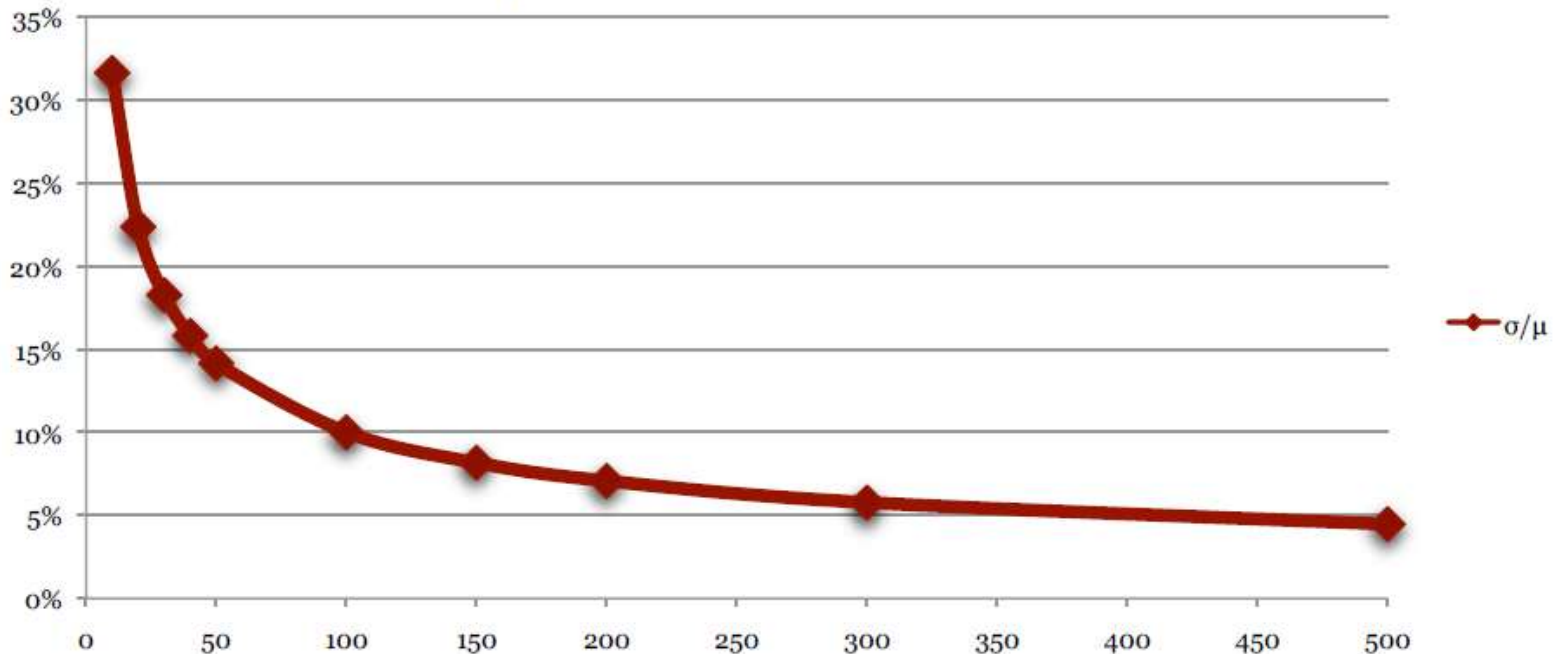
- Для суммы **независимых** случайных величин верно:  
$$\sigma = \sqrt{(\sigma_1^2 + \sigma_2^2 + \dots + \sigma_n^2)}$$
- Сигма суммы независимых случайных величин **уменьшается** при увеличении их количества:  
$$\frac{\sigma}{\mu} \xrightarrow{n \rightarrow \infty} 0$$
- Таким образом, чем больше в плане «независимых» задач, **тем точнее суммарная оценка сроков**. Погрешности планирования разных задач компенсируют друг друга.



# Управление сроками: сетевое планирование

## Зависимость сигмы от количества задач

- Показана зависимость общей сигмы плана в процентах от количества независимых задач.
- Задачи имеют равные длительности и сигму 100%.



## Какие задачи независимы?

- Независимых задач в разработке много:
  - Все задачи разработки, которые могут выполняться независимо друг от друга и впараллель;
  - Все задачи, относящиеся к непересекающемуся функционалу;
  - Большинство заданий, возникающих при поддержке ПО (исправление дефектов, реализация feature requests, и прочее).
- Прогнозы задач **зависимы**, если их реальные длительности зависят от одних и тех же факторов
  - Зависимые задачи обычно связаны связью «окончание-начало». Например, фазы разработки зависимы по прогнозу между собой.

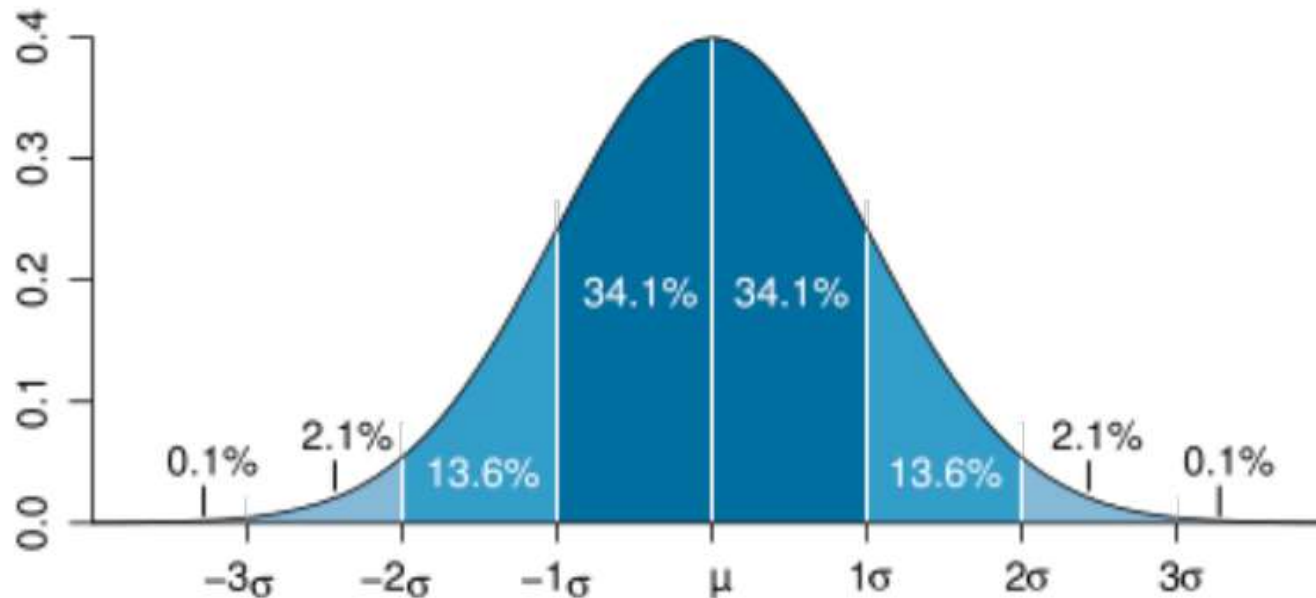
## Оценки для группы задач

- Для суммы случайных величин верно:  
$$\mu = \mu_1 + \mu_2 + \dots + \mu_n;$$
- Ожидаемое время выполнения задач просто суммируется.
- Сигма для группы задач:
  - Суммируется для зависимых прогнозов;
  - Может быть оценена как корень из суммы квадратов для независимых прогнозов.
- **Распределение суммы** случайных величин меняется, приближаясь к нормальному, при увеличении их количества.
- PERT: сумма задач уже не имеет бета-распределения.

# Управление сроками: сетевое планирование

## Нормальное распределение

- «Не справлюсь точно» (вероятность <2%)  
=  $\mu - 2\sigma$
- «Успею с запасом» (вероятность 98%)  
=  $\mu + 2\sigma$
- Между крайними оценками 4 сигмы





## PERT Estimation

- PERT Deviation лишен внятного смысла для суммы задач.
  - «Задача уложится в  $\mu + \sigma$  с вероятностью 72 %» - **для суммы задач уже не верно.**
  - Сколько сигм надо добавить к прогнозу сроков всего проекта, чтобы успеть с вероятностью 85% («скорее всего»)?
- PERT Estimation не лучше простой пары оценок «оптимистичная – пессимистичная»
  - Центральная оценка с весом 4 забивает крайние, и доминирует в прогнозе.
- В результате, PERT на практике не позволяет работать с большой неопределенностью в прогнозе.

# Управление сроками: сетевое планирование

## Модифицируем формулу PERT

- В предположении, что срок выполнения задачи имеет нормальное распределение:
  - «Не справлюсь точно» (вероятность <2%)  
 $= \mu - 2\sigma$
  - «Успею с запасом» (вероятность 98%)  
 $= \mu + 2\sigma$
- «Normal Estimation»:
  - $\mu = (L + H) / 2$   
 $\sigma = (H - L) / 4$
  - **Распределение сохраняется при суммировании.**
  - Проект уложится в срок  $\mu + \sigma$  с вероятностью  $\approx 85\%$ .

# Управление сроками: критический путь

Метод критического пути ([англ. CPM, Critical path method](#)) — инструмент [планирования](#) расписания и [управления сроками проекта](#)

В основе метода лежит определение наиболее длительной **последовательности задач** от начала [проекта](#) до его окончания с учетом их взаимосвязи

Задачи, лежащие на критическом пути (*критические задачи*), имеют **нулевой резерв** времени выполнения, и, в случае изменения их длительности, изменяются сроки всего проекта

В связи с этим, при выполнении проекта критические задачи требуют более тщательного контроля, в частности, своевременного **выявления проблем и рисков**, влияющих на сроки их выполнения и, следовательно, на сроки выполнения проекта в целом

В процессе выполнения проекта **критический путь проекта может меняться**, так как при изменении длительности задач некоторые из них могут оказаться на критическом пути.

# Критический путь: алгоритмы поиска

## Instructions

Click within the white grid and drag your mouse to draw obstacles.

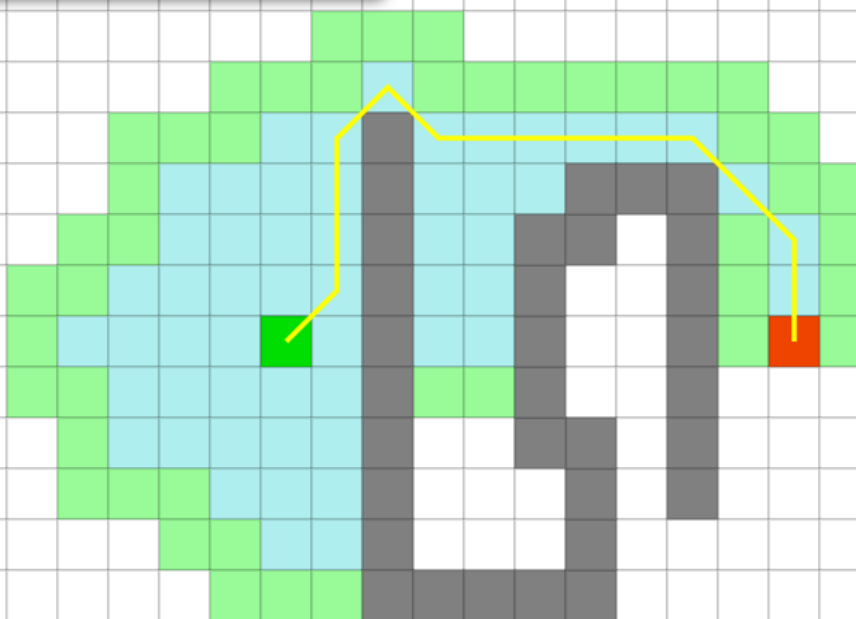
Drag the **green** node to set the start position.

Drag the red node to set the end position.

Choose an algorithm from the right-hand panel.

Click Start Search in the lower-right corner to start the animation.

hide



```
length: 17.07
time: 14.7500ms
operations: 157
```



# Планирование в agile: деловая игра

**Покер планирования** ([англ. Planning Poker](#), а также [англ. Scrum poker](#)) — техника оценки, основанная на достижении договорённости, главным образом используемая для оценки сложности предстоящей работы или относительного объёма решаемых задач при [разработке программного обеспечения](#)

Это разновидность метода [Wideband Delphi](#)

Она обычно используется в [гибкой методологии разработки](#), в частности, в [методологии экстремального программирования](#)

**Для проведения покера планирования необходимо:**

1. определить **роли участников**
2. определить **единицы измерения сложности**
3. подготовить **список обсуждаемых функций** ([пользовательских историй](#)), которые описывают разрабатываемое программное обеспечение
4. определить **временные ограничения** на обсуждение

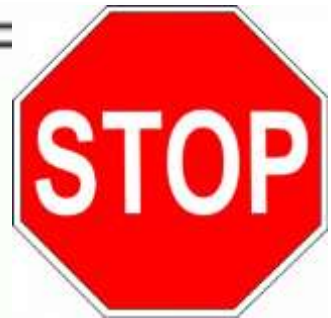
# Планирование в agile: совещания

**Ты одинок?**

Надоело работать самостоятельно?

Не любишь принимать решения?

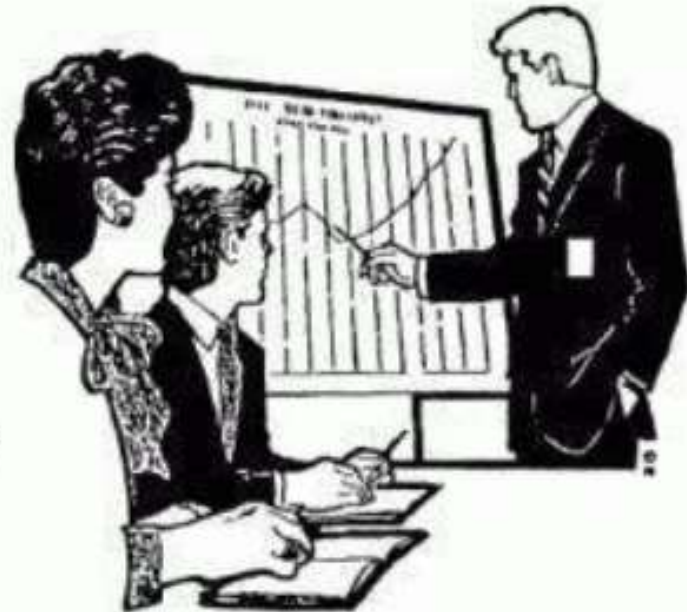
**СОБЕРИ СОВЕЩАНИЕ!**



Ты сможешь:

- посмотреть на других
- показать графики
- почувствовать себя важным
- поддержать указку
- перекусить
- произвести впечатление на коллег

И все это в рабочее время!



**СОВЕЩАНИЯ**

**РЕАЛЬНАЯ АЛЬТЕРНАТИВА РАБОТЕ**

океан

# Планирование: личный тайм-менеджмент

**Управление временем, организация времени, тайм-менеджмент** ([англ. \*time management\*](#)) — технология организации времени и повышения эффективности его использования

Управление временем — это **действие** или **процесс** тренировки **сознательного контроля над временем**, потраченным на конкретные виды деятельности, при котором специально увеличиваются эффективность и продуктивность

Управление временем может помочь получить ряд **навыков, инструментов и методов**, используемых при выполнении конкретных задач:

- планирование,
- распределение,
- постановку целей,
- делегирование,
- анализ временных затрат,
- мониторинг,
- организацию,
- составление списков и расстановку приоритетов

# Планирование: личный тайм-менеджмент

## Планирование «сегодня»

- ☐ **Жесткие задачи** – задачи, которые должны быть выполнены к определенному времени
- ☐ **Мягкие задачи** – задачи, которые нужно сделать, но жесткого срока нет
- ☐ **Кайросы** – задачи, которые нужно делать «когда поперло»

## «Слоны», «лягушки» и «конфетки»

- ☐ **Слон** – большая задача, которую уместить в голову целиком невозможно
  - декомпозировать на более мелкие, делать по частям
  - начинать с того, что понятно, постепенно «связывая» куски
- ☐ **Лягушка** – неприятная задача, которую не хочется делать
  - вести список «лягушек»
  - каждый день есть по крайней мере одну
- ☐ **Конфетка** – приятная задача или «бонус»
  - Оставлять на самый конец
  - При несъедании лягушки лишать себя бонуса



# Планирование: личный тайм-менеджмент

## Качество рабочего времени - поток

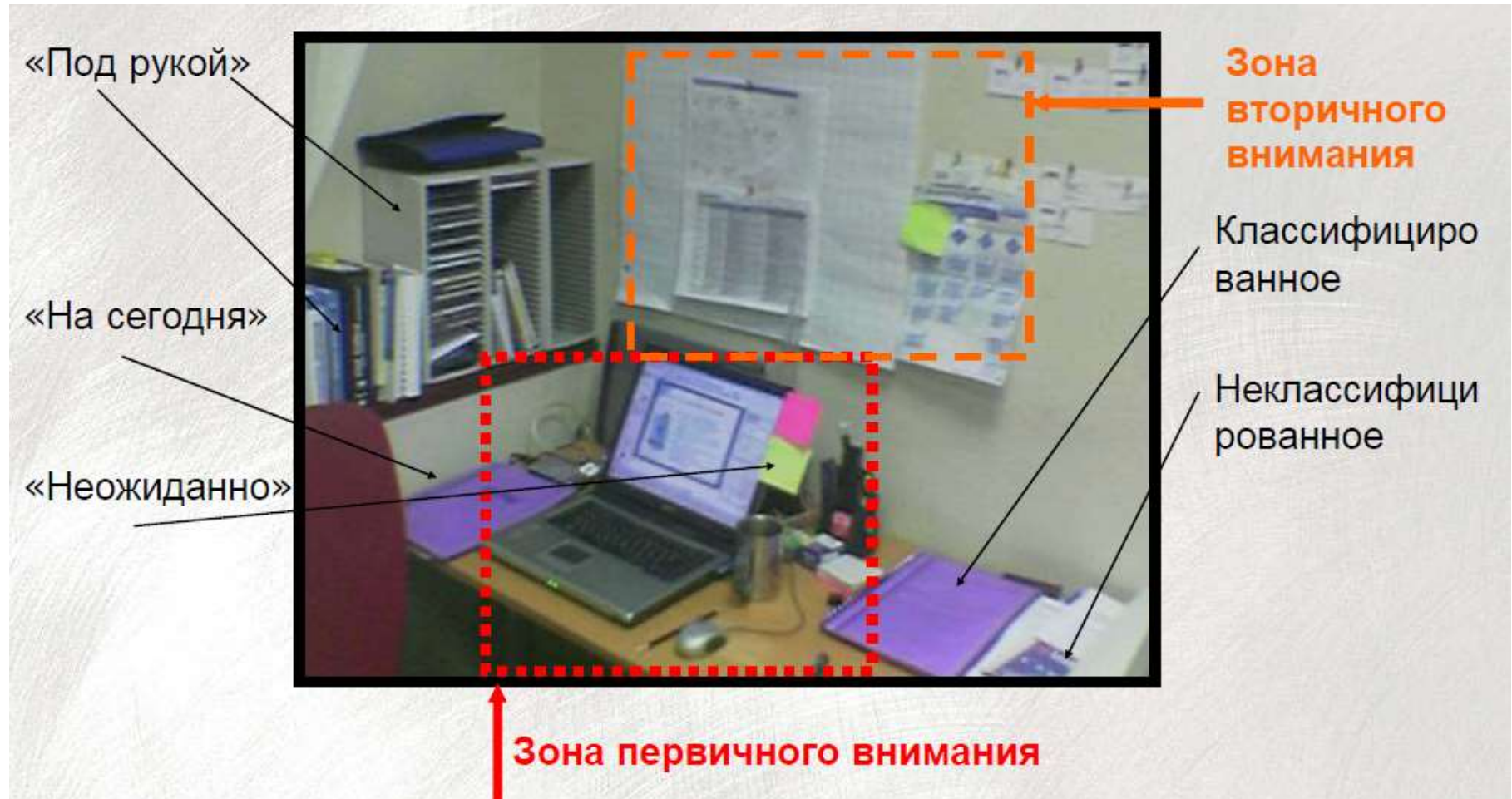
- ☐ Когда Вы выполняете задачу, Вы выполняете задачу. Это значит:
  - Вы не говорите по телефону (выключите или активируйте «белый лист»)
  - Не отвечаете на email (кроме суперкритичного)
  - Не отвлекаетесь на разговоры (кроме абсолютно необходимых)

## Качество рабочего времени - ритм

- ☐ Ритм
  - на один час 5 минут отдыха
- ☐ Смена занятия
  - читал – смотри вдаль
  - сидел – покрути головой, сделай наклонов
- ☐ Приятное и полезное
  - Вытащить всех на stand-up meeting за кофе
  - Спуститься пообщаться с клиентом

# Планирование: личный тайм-менеджмент

## Качество рабочего пространства





# Планирование: личный тайм-менеджмент

**РАБОТАТЬ НУЖНО**  
не 12 часов, а головой!

\ Стив Джобс \

3 + 6 = 5

