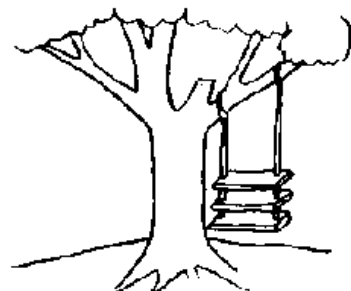
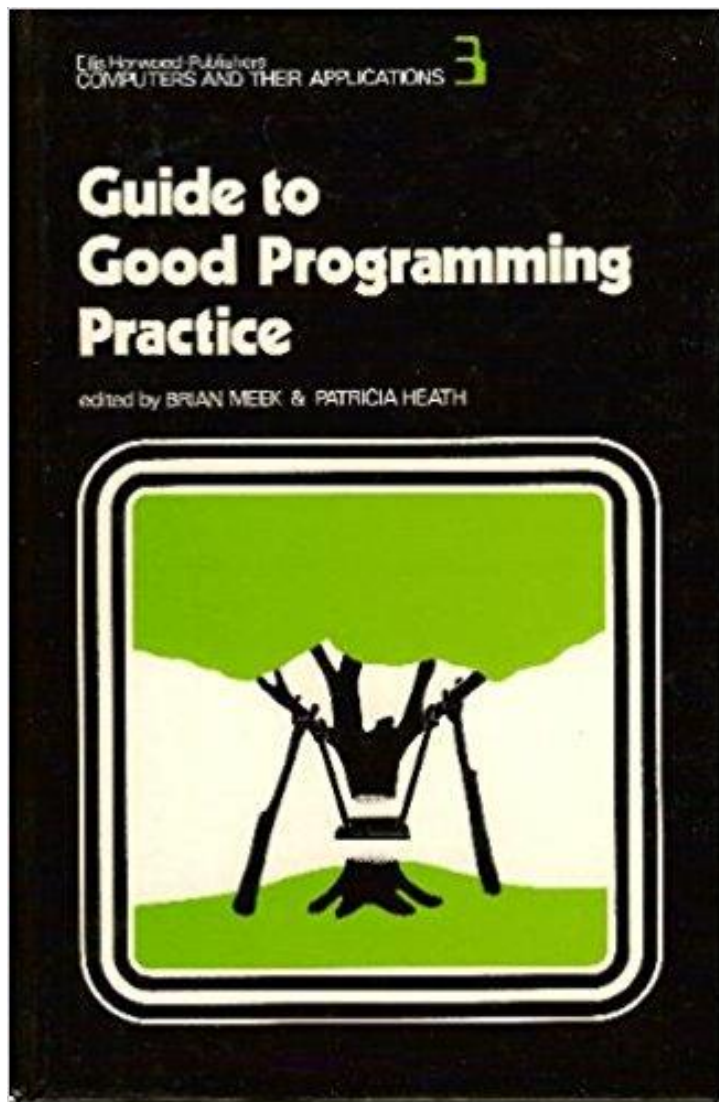


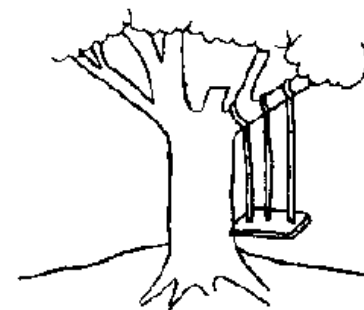
Лекция 5
**«Методологии
проектирования ИС»**
Обзор

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

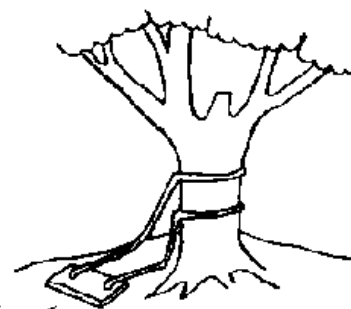
Проблематика



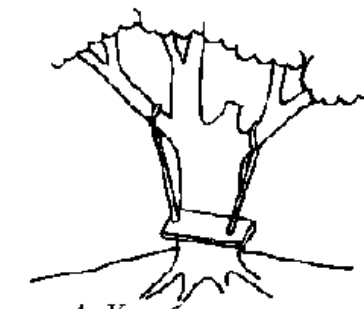
1. Как было предложено организатором разработки



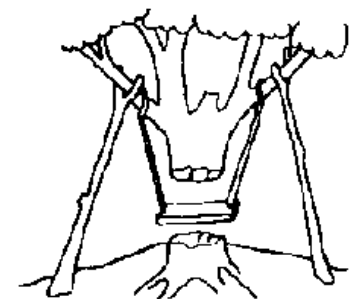
2. Как было описано в техническом задании



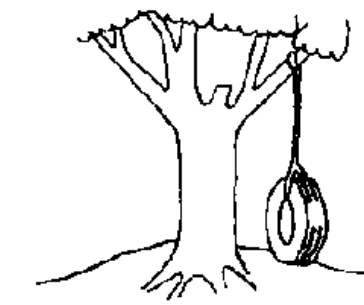
3. Как было спроектировано ведущим системным специалистом



4. Как было реализовано программистами

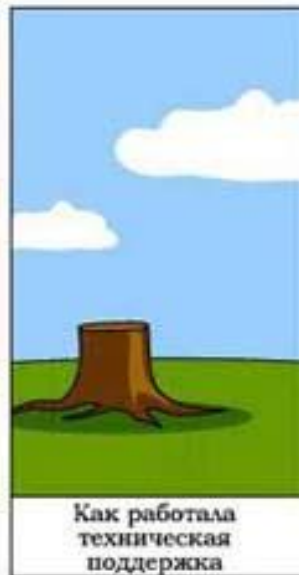
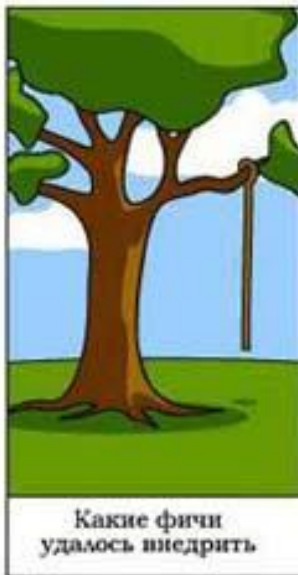


5. Как было внедрено



6. Чего хотел пользователь

Проблематика



Терминология: методология

Мéтод (от др.-греч. μέθοδος — путь исследования или познания, от μετά- + ὁδός «путь») — систематизированная совокупность шагов, действий, которые нацелены на решение определённой задачи или достижение определённой цели.

В отличие от области знаний или исследований, является авторским, то есть созданным конкретной персоной или группой персон, научной или практической школой. В силу своей ограниченности рамками действия и результата, методы имеют тенденцию устаревать, преобразовываясь в другие методы, развиваясь в соответствии со временем, достижениями технической и научной мысли, потребностями общества. Совокупность однородных методов принято называть подходом. Развитие методов является естественным следствием развития научной мысли.

Алгоритм — набор инструкций, описывающих порядок действий исполнителя для достижения некоторого результата.

В старой трактовке вместо слова «порядок» использовалось слово «последовательность», но по мере развития параллельности в работе компьютеров слово «последовательность» стали заменять более общим словом «порядок».

Независимые инструкции могут выполняться в произвольном порядке, параллельно, если это позволяют используемые исполнители.

Часто в качестве исполнителя выступает компьютер, но понятие алгоритма необязательно относится к компьютерным программам, так, например, чётко описанный рецепт приготовления блюда также является алгоритмом, в таком случае исполнителем является человек (а может быть и некоторый механизм, ткацкий станок, и пр.).

[Метод \(Википедия\)](#)

[Алгоритм \(Википедия\)](#)

Терминология: методология

Методоло́гия (от [греч.](#) μεθοδολογία — учение о [способах](#); от [др.-греч.](#) μέθοδος из [μετά-](#) + ὁδός, букв. «путь вслед за чем-либо» и [др.-греч.](#) λόγος — [мысль](#), [причина](#)) — учение о [методах](#), способах и стратегиях исследования предмета.

В методологии можно выделить следующую структуру:

- основания методологии: [философия](#), [логика](#), [системология](#), [психология](#), [информатика](#), [системный анализ](#), [науковедение](#), [этика](#), [эстетика](#);
- характеристики деятельности: особенности, принципы, условия, нормы деятельности;
- логическая структура деятельности: [субъект](#), [объект](#), [предмет](#), [формы](#), [средства](#), [методы](#), результат деятельности, [решение задач](#);
- временная структура деятельности: фазы, стадии, этапы.
- [технология](#) выполнения работ и решения задач: средства, методы, способы, приемы.

Теория управлѐния — [наука](#) о принципах и методах управления различными системами, процессами и объектами.

Теоретической базой теории управления являются [кибернетика](#) и [теория информации](#).

Суть теории управления состоит в построении на основе [анализа](#) данной системы, процесса или объекта такой [абстрактной модели](#), который позволит получить [алгоритм](#) управления ими в динамике, — для достижения системой, процессом или объектом состояния, которое требуется целями управления.

[Методология \(Википедия\)](#)

[Теория управления \(Википедия\)](#)

Методология ГОСТ 34

ГОСТ 34.601-90 Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания

1. Формирование требований к АС
2. Разработка концепции АС
3. Техническое задание
4. Эскизный проект
5. Технический проект
6. Рабочая документация
7. Ввод в действие
8. Сопровождение АС

РД 50-34.698-90 Методические указания. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов

1.1. Требования к содержанию документов, разрабатываемых при создании АС, установлены настоящими указаниями, а также соответствующими государственными стандартами Единой системы программной документации (ЕСПД), Единой системы конструкторской документации (ЕСКД), Системы проектной документации для строительства (СПДС) и [ГОСТ 34.602](#).

Виды и комплектность документов регламентированы [ГОСТ 34.201](#).

[ГОСТ 34.601-90](#)

[РД 50-34.698-90](#)

Методология ГОСТ Р ИСО/МЭК 12207-2010

ГОСТ Р ИСО/МЭК 12207-2010 Информационная технология (ИТ). Системная и программная инженерия. Процессы жизненного цикла программных средств

5.1.12 Модели и стадии жизненного цикла

Процесс жизни любой системы или программного продукта может быть описан посредством модели жизненного цикла, состоящей из стадий. Модели могут использоваться для представления всего жизненного цикла от замысла до прекращения применения или для представления части жизненного цикла, соответствующей текущему проекту. Модель жизненного цикла представляется в виде последовательности стадий, которые могут перекрываться и (или) повторяться циклически в соответствии с областью применения, размером, сложностью, потребностью в изменениях и возможностях.

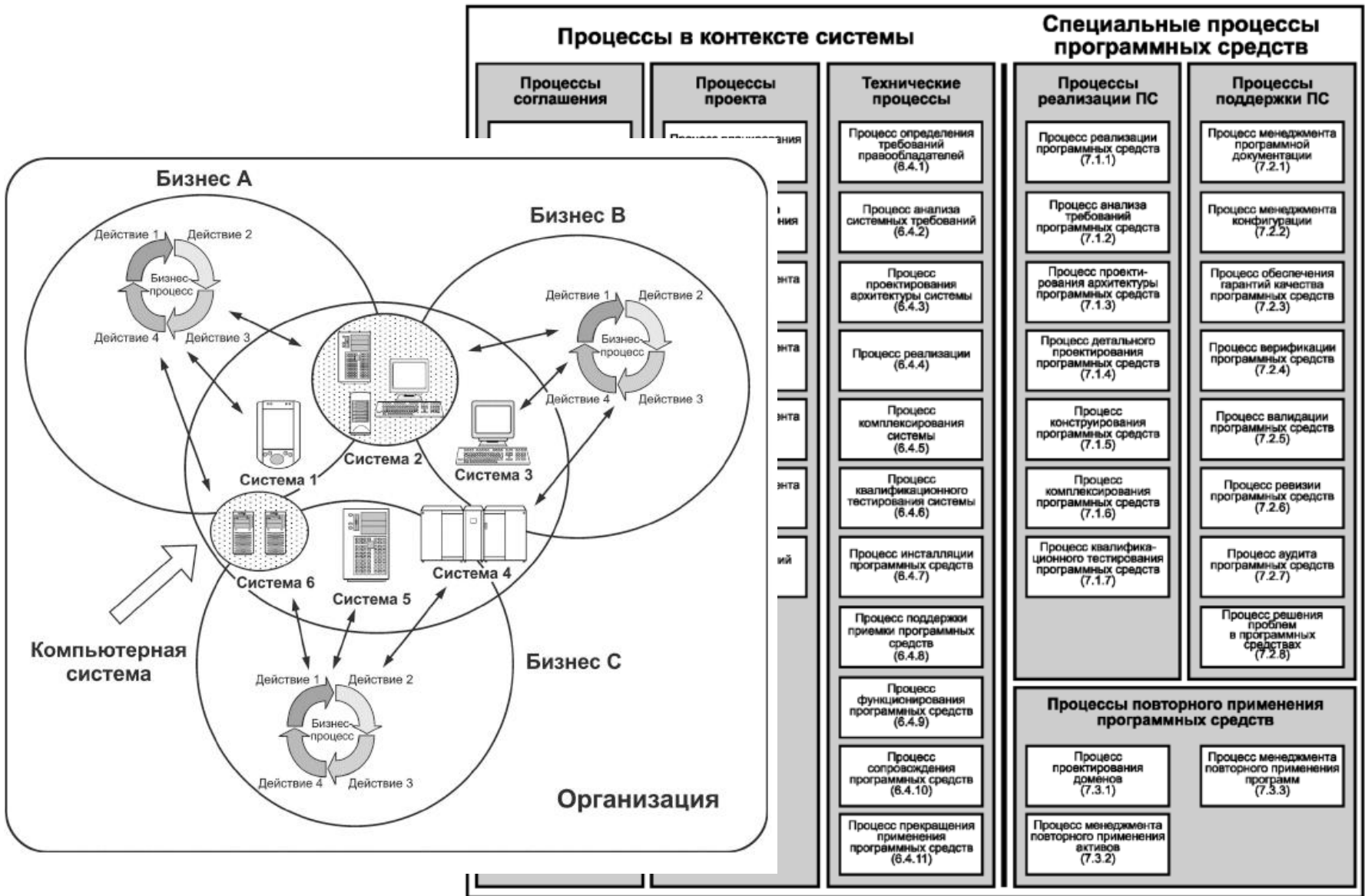
5.2.1 Категории процессов жизненного цикла

Настоящий стандарт группирует различные виды деятельности, которые могут выполняться в течение жизненного цикла программных систем, в семь групп процессов. Каждый из процессов жизненного цикла в пределах этих групп описывается в терминах цели и желаемых выходов, списков действий и задач, которые необходимо выполнять для достижения этих результатов.

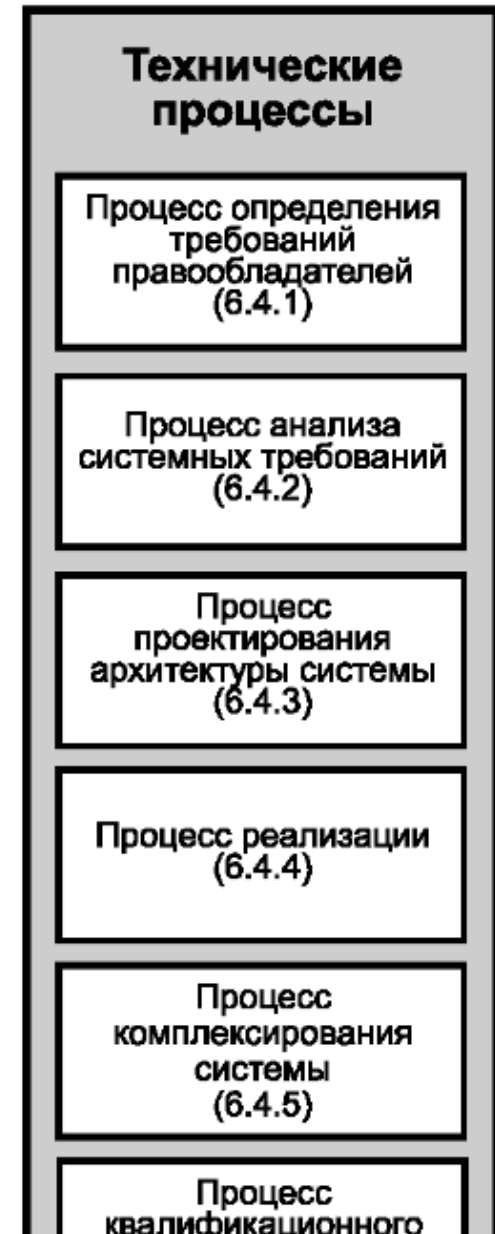
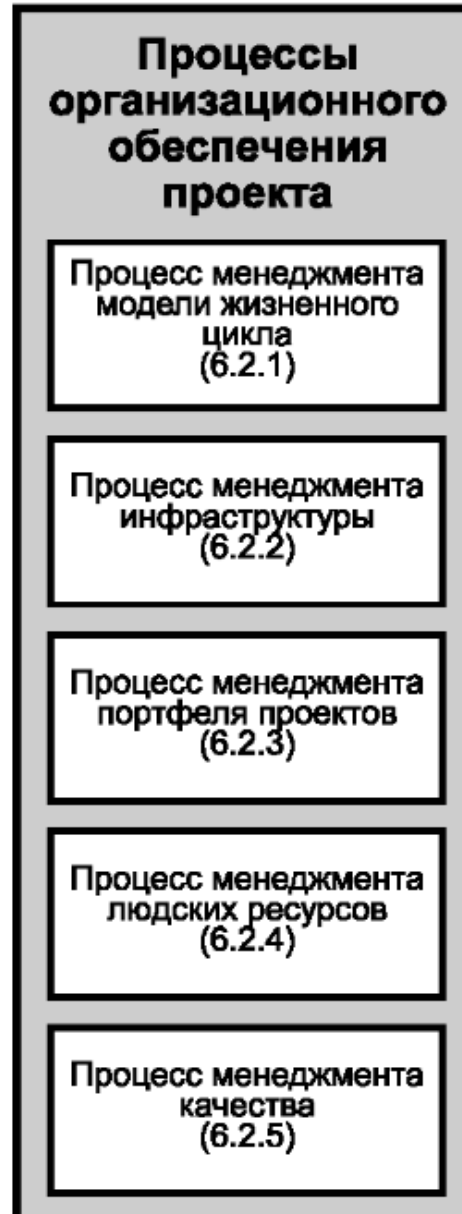
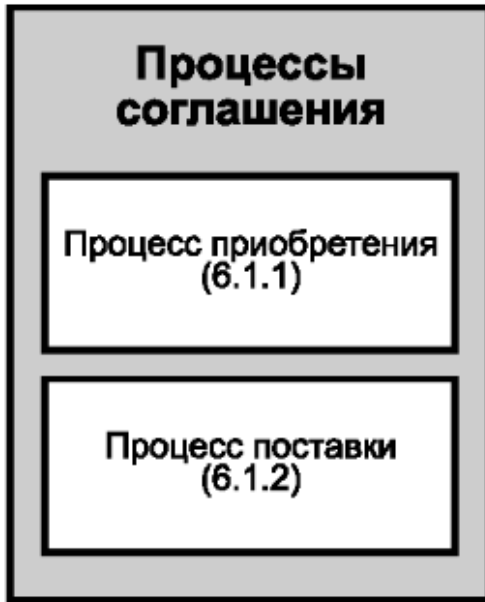
- a) процессы **соглашения** - два процесса (см. 5.2.2.1.1 и 6.1);
- b) процессы **организационного обеспечения проекта** - пять процессов (см. 5.2.2.1.2 и 6.2);
- c) процессы **проекта** - семь процессов (см. 5.2.2.1.3 и 6.3);
- d) **технические** процессы - одиннадцать процессов (см. 5.2.2.1.4 и 6.4);
- e) процессы **реализации программных средств** - семь процессов (см. 5.2.2.2.1 и 7.1);
- f) процессы **поддержки программных средств** - восемь процессов (см. 5.2.2.2.2 и 7.2);
- g) процессы **повторного применения программных средств** - три процесса (см. 5.2.2.2.3 и 7.3).

Цели и результаты процессов жизненного цикла образуют эталонную модель процессов.

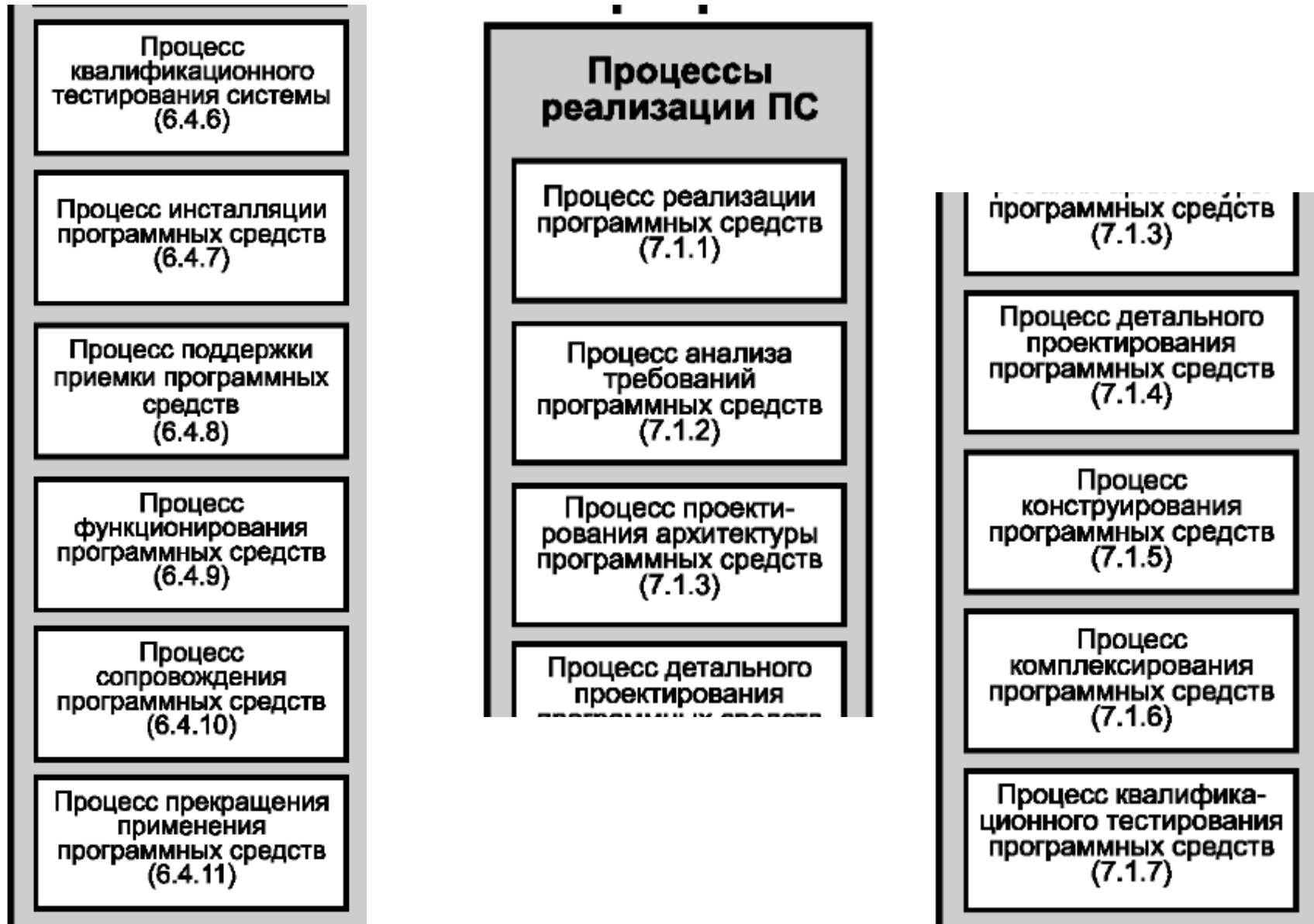
Методология ГОСТ Р ИСО/МЭК 12207-2010



Методология ГОСТ Р ИСО/МЭК 12207-2010



Методология ГОСТ Р ИСО/МЭК 12207-2010



Методология ГОСТ Р ИСО/МЭК 12207-2010



Методология CMMI

Capability Maturity Model Integration (CMMI) — набор моделей (методологий) совершенствования процессов в организациях разных размеров и видов деятельности. CMMI содержит набор рекомендаций в виде практик, реализация которых, по мнению разработчиков модели, позволяет реализовать цели, необходимые для полной реализации определённых областей деятельности.

Набор моделей CMMI включает три модели:
CMMI for Development (CMMI-DEV),
CMMI for Services (CMMI-SVC) и
CMMI for Acquisition (CMMI-ACQ).

Наиболее известной является модель CMMI for Development, ориентированная на организации, занимающиеся разработкой программного обеспечения, аппаратного обеспечения, а также комплексных систем

Концепция зрелости процессов



Методология RUP

Rational Unified Process (RUP) — [методология](#) разработки программного обеспечения, созданная компанией [Rational Software](#).

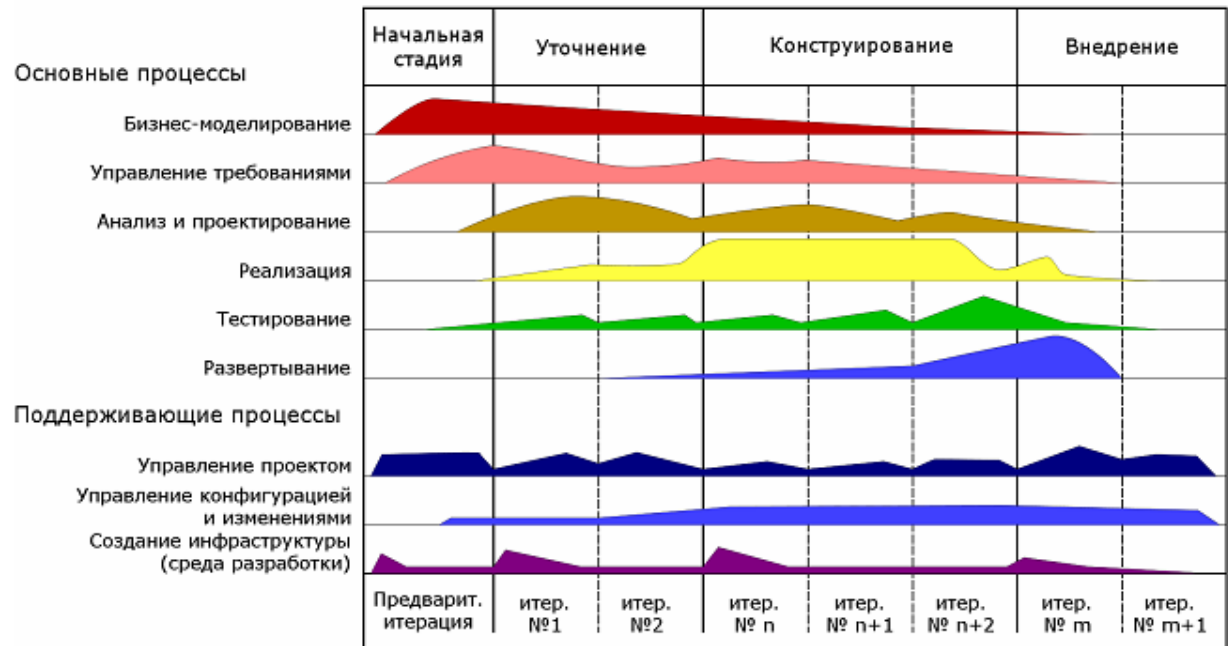
В основе RUP лежат следующие принципы:

- Ранняя идентификация и непрерывное (до окончания [проекта](#)) устранение основных рисков.
- Концентрация на выполнении требований заказчиков к исполняемой программе (анализ и построение модели [прецедентов](#) (вариантов использования)).
- Ожидание изменений в требованиях, проектных решениях и реализации в процессе разработки.
- [Компонентная архитектура](#), реализуемая и тестируемая на ранних стадиях проекта.
- Постоянное обеспечение качества на всех этапах разработки [проекта](#) (продукта).
- Работа над проектом в сплочённой команде, ключевая роль в которой принадлежит архитекторам.

1996 – старт
 1999 – UML 1.3
 2003 – IBM
 2006 - [OpenUP](#)

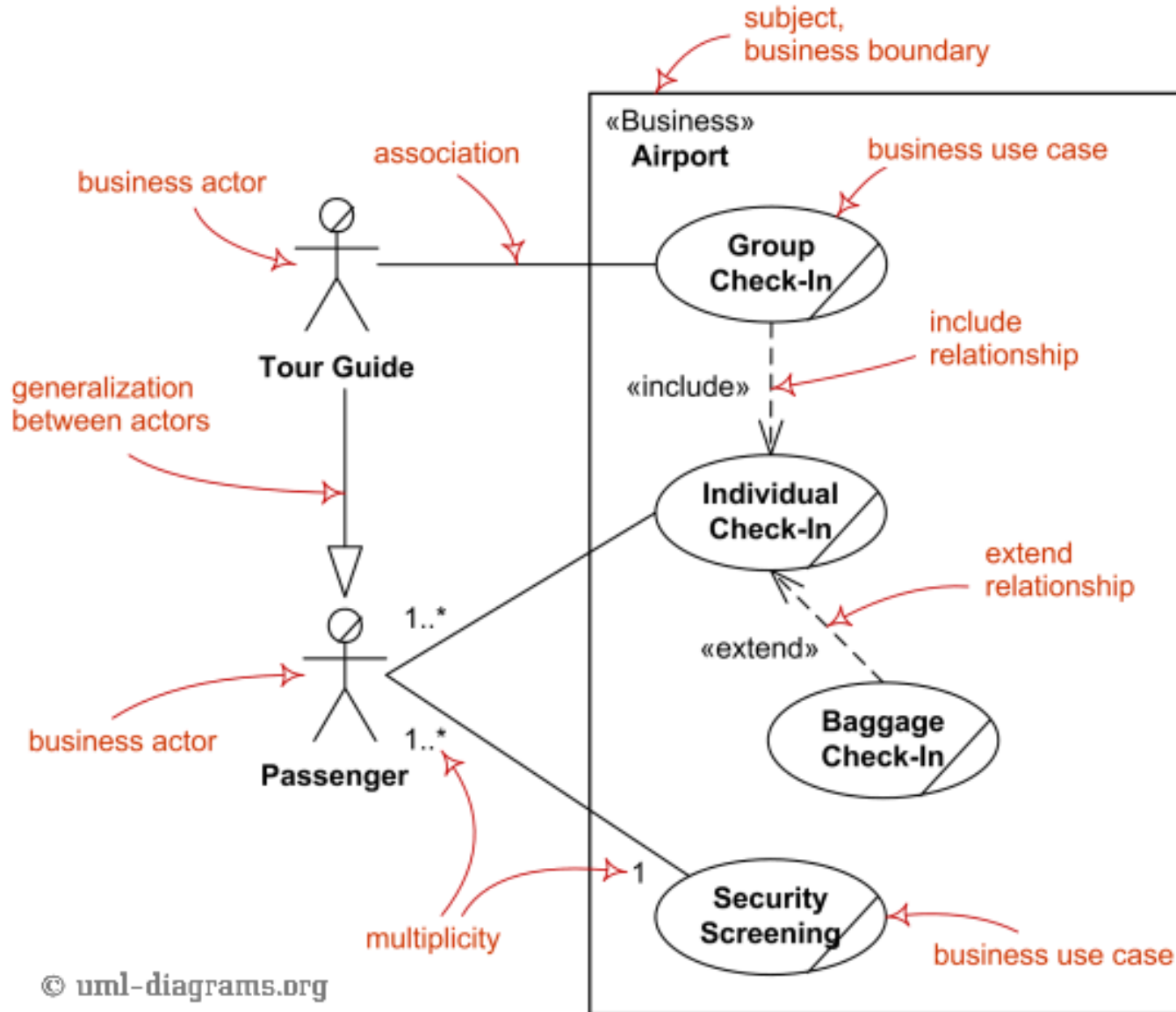
Рабочие процессы

Стадии

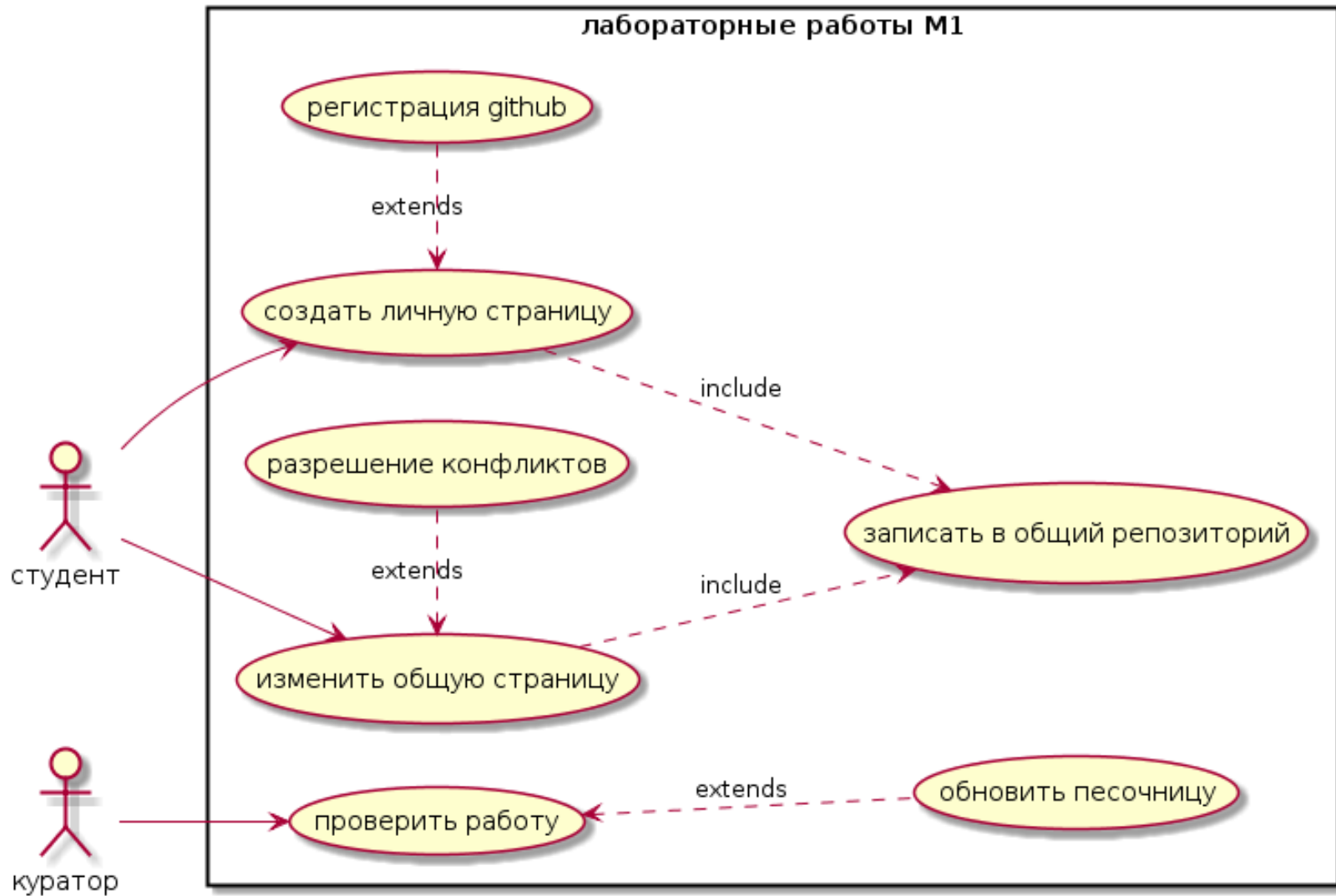


Итерации

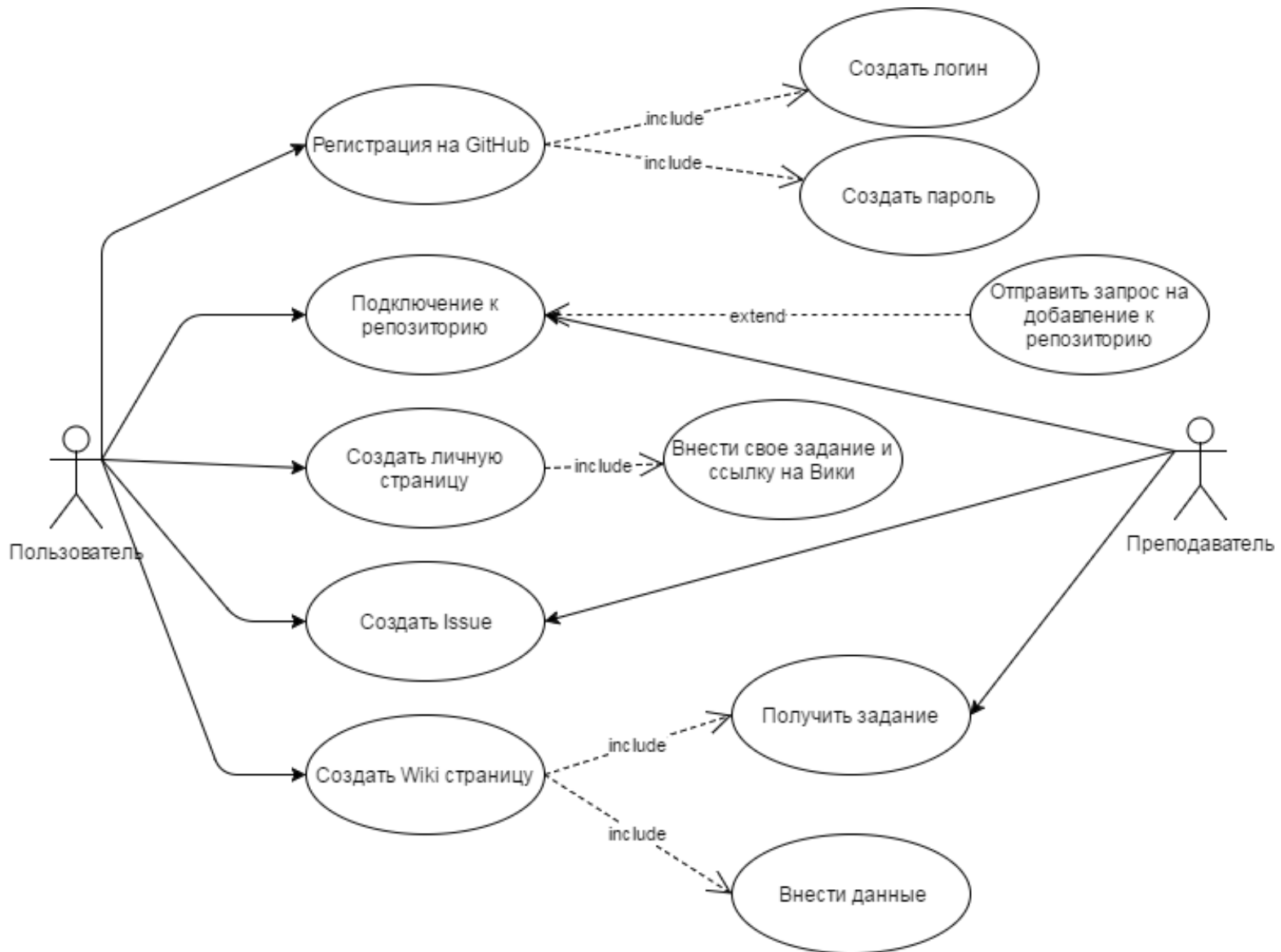
Методология RUP



Методология RUP



Методология RUP



Методологии Agile

Гибкая методология разработки ([англ. Agile software development, agile-методы](#)) — серия подходов к [разработке программного обеспечения](#), ориентированных на использование [итеративной](#) разработки, динамическое формирование требований и обеспечение их реализации в результате постоянного взаимодействия внутри самоорганизующихся рабочих групп, состоящих из специалистов различного профиля^[1].

Существует несколько методик, относящихся к классу гибких методологий разработки, в частности [экстремальное программирование](#), [DSDM](#), [Scrum](#), [FDD](#)..

- ❑ **Люди и взаимодействия** важнее чем процессы и инструменты
- ❑ **Работающий код** важнее совершенной документации
- ❑ **Сотрудничество с заказчиком** важнее контрактных обязательств
- ❑ **Реакция на изменения** важнее следования плану

Методология XP

Экстремальное программирование ([англ. Extreme Programming, XP](#)) — одна из [гибких методологий разработки программного обеспечения](#).

Двенадцать основных приёмов экстремального программирования (по первому изданию книги *Extreme programming explained*) могут быть объединены в четыре группы:

Короткий цикл обратной связи (Fine-scale feedback)

[Разработка через тестирование](#) (Test-driven development)

Игра в планирование (Planning game)

Заказчик всегда рядом (Whole team, Onsite customer)

[Парное программирование](#) (Pair programming)

Непрерывный, а не пакетный процесс

[Непрерывная интеграция](#) (Continuous integration)

[Рефакторинг](#) (Design improvement, Refactoring)

Частые небольшие релизы (Small releases)

Понимание, разделяемое всеми

Простота проектирования (Simple design)

Метафора системы

Коллективное владение кодом (Collective code ownership) или выбранными шаблонами проектирования (Collective patterns ownership)

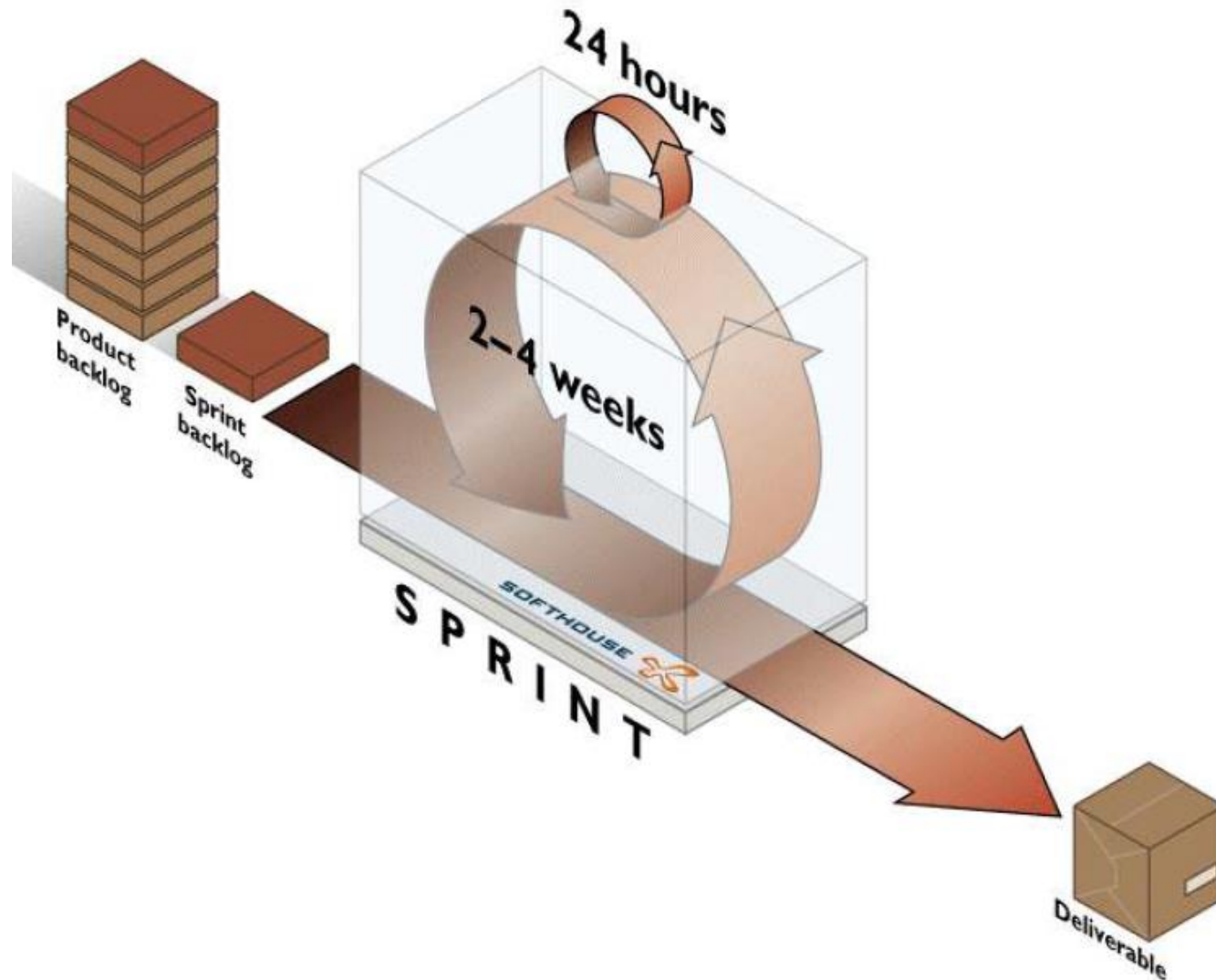
[Стандарт оформления кода](#) (Coding standard or Coding conventions)

Социальная защищённость программиста (Programmer welfare):

40-часовая рабочая неделя (Sustainable pace, Forty-hour week)

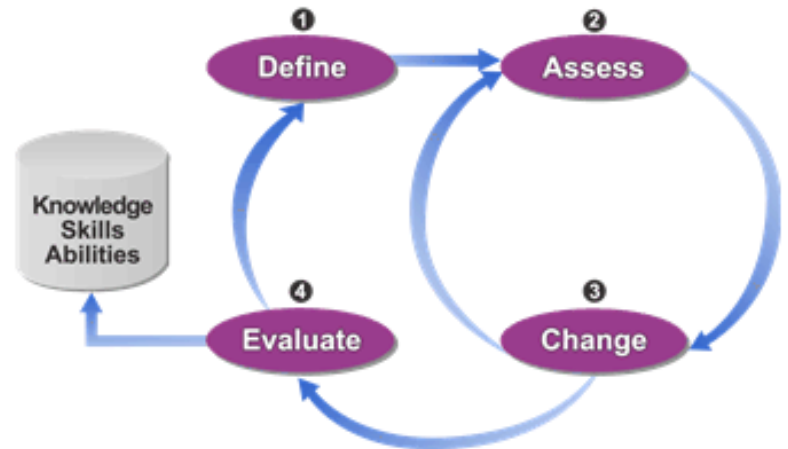
Методология Scrum

Scrum ([/skrʌm/\[1\]\[2\]](#); [англ. scrum](#) «схватка») — методология [гибкой разработки](#) ПО. Методология делает акцент на качественном контроле процесса разработки.



Методология MSF

Microsoft Solutions Framework (MSF) — методология разработки программного обеспечения, предложенная корпорацией Microsoft. MSF опирается на практический опыт Microsoft и описывает управление людьми и рабочими процессами в процессе разработки решения.



Методология OUM

Oracle Unified Method (унифицированный метод Oracle, сокр. OUM) - фреймворк для итеративного и инкрементального процесса разработки ПО, разработанный корпорацией Oracle для реализации своей точки зрения на поддержку успешной реализации каждого продукта Oracle – приложений, промежуточного ПО и баз данных.

Процессы разработки

RD - Определение производственных требований,

ES - Исследование существующих систем,

TA - Определение технической архитектуры,

DB - Проектирование и построение БД,

MD - Проектирование и реализация модулей,

CV - Конвертирование данных,

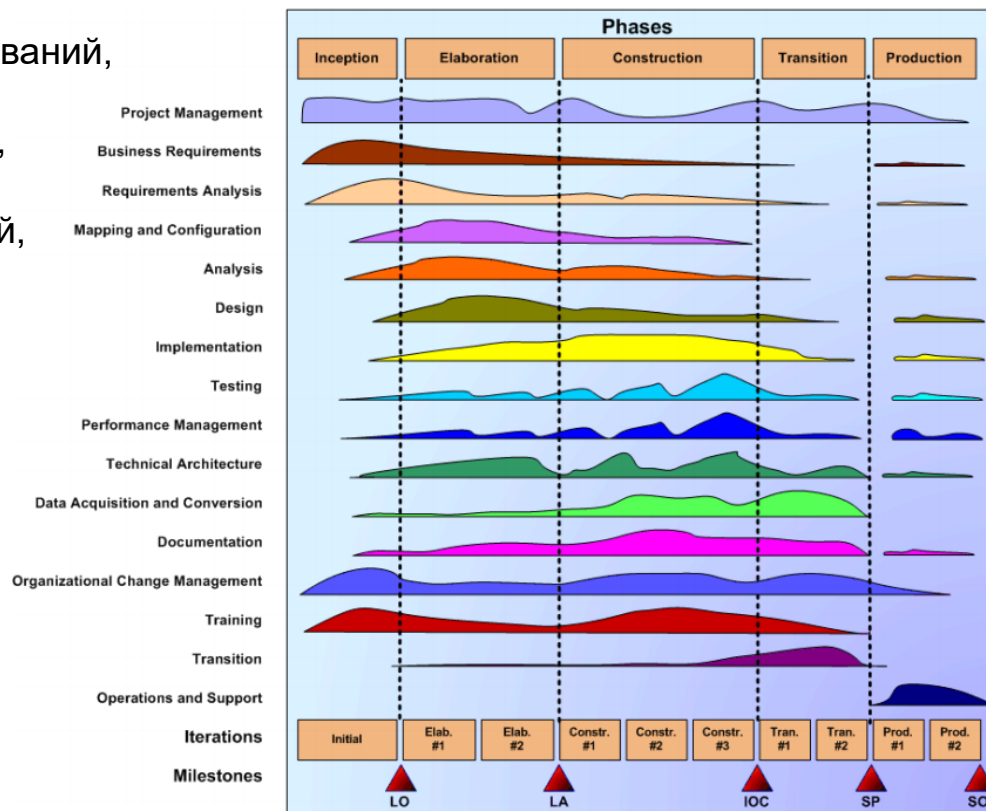
DO - Документирование,

TE - Тестирование,

TR - Обучение,

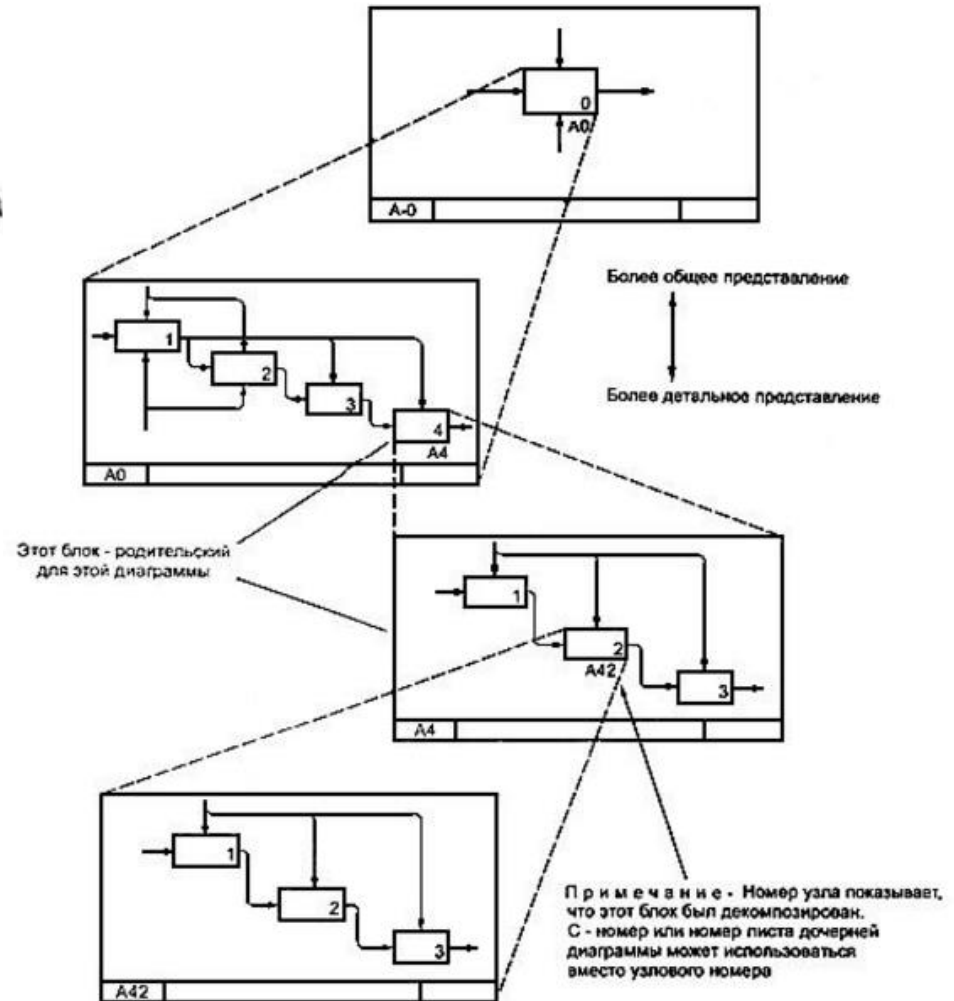
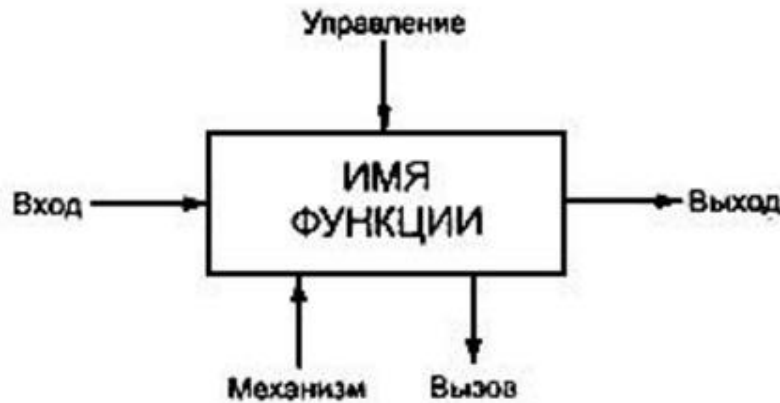
TS - Переход к новой системе,

PS - Поддержка и сопровождение.



Методология SADT

Р 50.1.028-2001 Информационные технологии поддержки жизненного цикла продукции. Методология функционального моделирования



Методология DFD

DFD — общепринятое сокращение от [англ. data flow diagrams](#) — диаграммы потоков данных. Так называется [методология](#) графического структурного [анализа](#), описывающая внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ.



Лекция 6
«Методологии проектирования ИС»
Проектирование данных и взаимодействия

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

Проектирование БД

ER-модель (от англ. *entity-relationship model*, модель «сущность — связь») — модель данных, позволяющая описывать концептуальные схемы предметной области.

ER-модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

Во время проектирования баз данных происходит преобразование ER-модели в конкретную схему базы данных на основе выбранной модели данных (реляционной, объектной, сетевой или др.).

Реляционная модель данных (РМД) — логическая модель данных, прикладная теория построения баз данных, которая является приложением к задачам обработки данных таких разделов математики, как теория множеств и логика первого порядка.

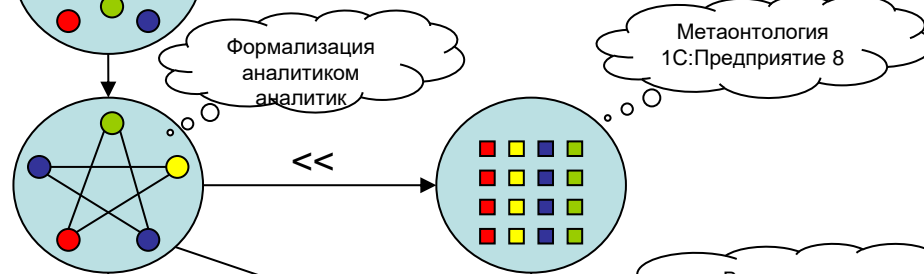
На реляционной модели данных строятся реляционные базы данных.

Проектирование БД

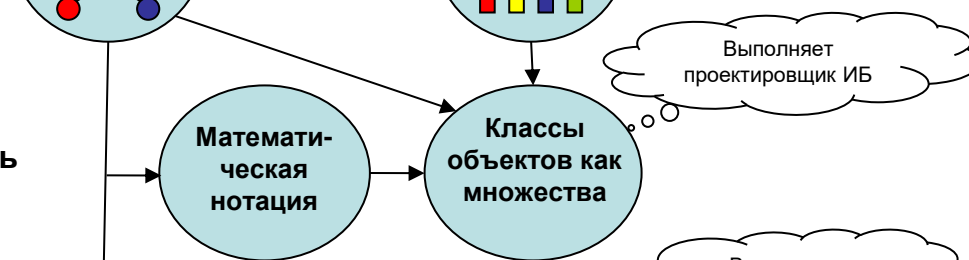
1. Семантическая сеть



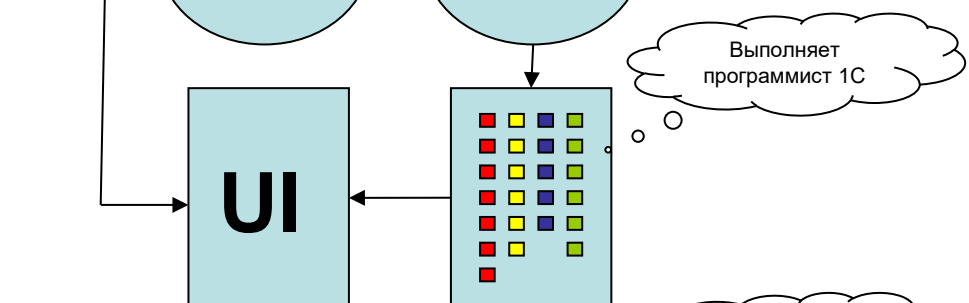
2. Онтология



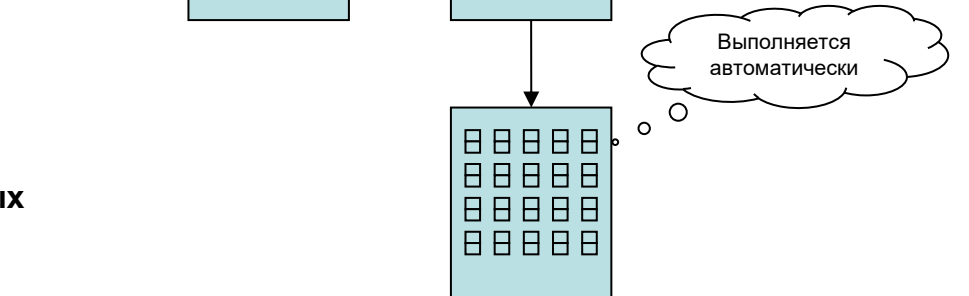
3. Концептуальная модель



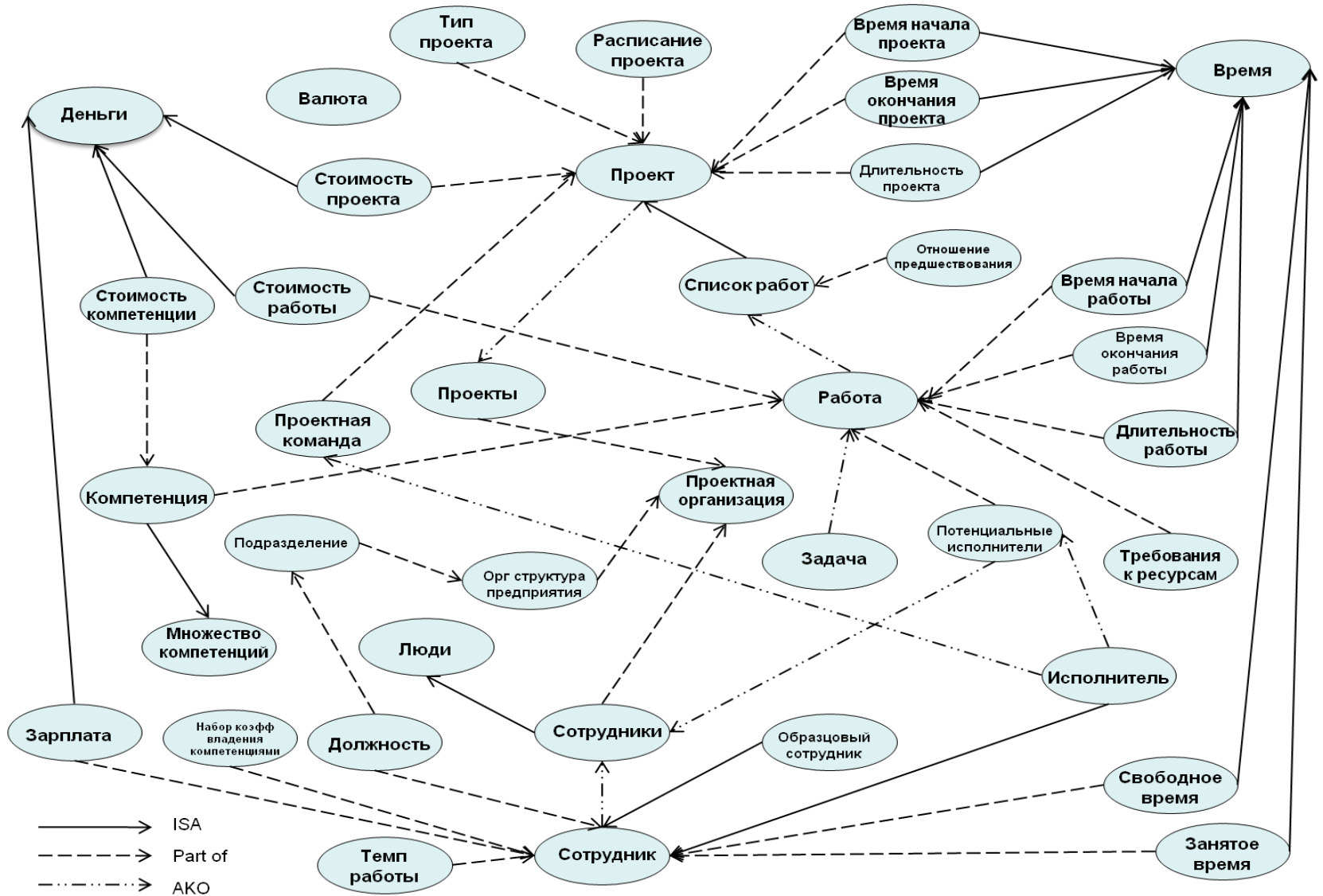
4. Объектная модель информационной базы



5. Логическая модель реляционной базы данных



Проектирование БД



Проектирование БД

OLAP ([англ. online analytical processing](#), интерактивная аналитическая обработка) — технология обработки данных, заключающаяся в подготовке суммарной (агрегированной) информации на основе больших массивов данных, структурированных по многомерному принципу. Реализации технологии OLAP являются компонентами программных решений класса [Business Intelligence](#)^[1].

Существуют три типа OLAP:^[2]

многомерная OLAP (Multidimensional OLAP — [MOLAP](#));

реляционная OLAP (Relational OLAP — [ROLAP](#));

гибридная OLAP (Hybrid OLAP — [HOLAP](#)).

NoSQL ([англ. not only SQL, не только SQL](#)), в [информатике](#) — термин, обозначающий ряд подходов, направленных на реализацию хранилищ [баз данных](#), имеющих существенные отличия от моделей, используемых в традиционных [реляционных СУБД](#) с доступом к данным средствами языка [SQL](#).

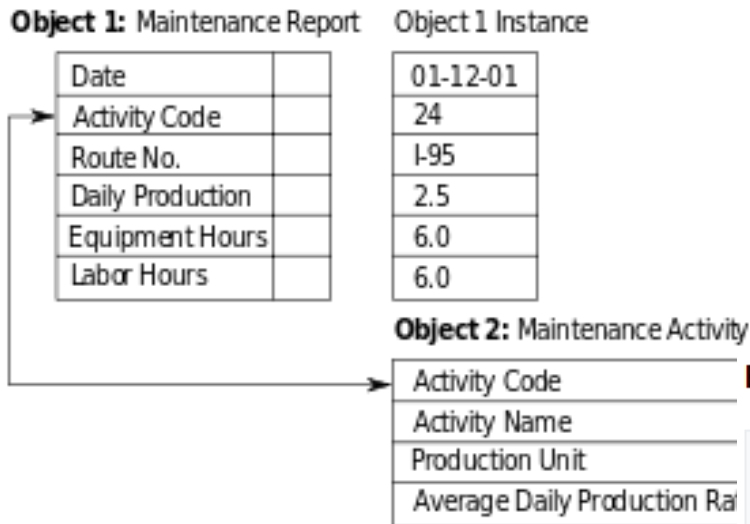
Применяется к базам данных, в которых делается попытка решить проблемы [масштабируемости](#) ([англ. scalability](#)) и [доступности](#) ([англ. availability](#)) за счёт [атомарности](#) ([англ. atomicity](#)) и [согласованности данных](#) ([англ. consistency](#))^[1].

Проектирование БД

Объектно-реляционная СУБД (ОРСУБД) — [реляционная СУБД \(РСУБД\)](#), поддерживающая некоторые технологии, реализующие [объектно-ориентированный подход](#): объекты, классы и наследование реализованы в структуре баз данных и языке запросов.

Объектно-реляционными [СУБД](#) являются, например, широко известные [Oracle Database](#), [Informix](#), [DB2](#), [PostgreSQL](#).

Object-Oriented Model



Example of XML Type Query in IBM DB2 SQL [\[edit\]](#)

```
select
  id, vol, xmlquery('$j/name', passing journal as "j") as name
from
  journals
where
  xmlexists('$j[licence="CreativeCommons"]', passing journal as "j")
```

Транзакции (экономика)

Экономика: Теория транзакционных издержек

Теория транзакционных издержек ([англ.](#) transaction cost theory) считается составной частью [Новой институциональной теории](#) и представляет собой теорию организации предприятий, объектом изучения которой служит многосторонний договор как форма организации.

Задачей теории транзакционных издержек является объяснение проблем эффективности тех или иных экономических операций в определённых институциональных рамках, то есть способность различных организационных форм в результативном планировании и осуществлении экономических целей. В основе данной теории находится предположение, что любое действие в экономическом контексте в первую очередь связано с затратами.

Транзакционные издержки — [затраты](#), возникающие в связи с заключением контрактов (в том числе с использованием рыночных механизмов); издержки, сопровождающие взаимоотношения экономических агентов. Выделяют следующие издержки:

- сбора и обработки информации
- проведения переговоров и принятия решений
- контроля
- юридической защиты выполнения

Транзакции (экономика)

Стратегическое управление: Теорема Коуза и Конфигурации Минцберга



Транзакции (психология)

Трансакция — это единица общения, которая состоит из стимула и реакции.

Трансакционный анализ (*транзакционный анализ, транзактный анализ, транзактный анализ*; сокр. ТА) представляет собой психологическую модель, служащую для описания и анализа поведения человека как индивидуально, так и в составе групп. Данная модель включает философию, теорию и методы, позволяющие людям понять самих себя и особенность своего взаимодействия с окружающими.

Краеугольным камнем транзакционного анализа является положение о том, что один и тот же человек, находясь в некоей определённой ситуации, может функционировать, исходя из одного из трёх эго-состояний, чётко отличимых одно от другого:

Эго-состояние Родителя (Р) содержит установки и поведение, перенятые извне, в первую очередь — от родителей. Внешне они часто выражаются в предубеждениях, критическом и заботливом поведении по отношению к другим. Внутренне — переживаются как старые родительские назидания, которые продолжают влиять на нашего внутреннего Ребёнка.

Эго-состояние Взрослого (В) не зависит от возраста личности. Оно ориентировано на восприятие текущей реальности и на получение объективной информации. Оно является организованным, хорошо приспособленным, находчивым и действует, изучая реальность, оценивая свои возможности и спокойно рассчитывая.

Эго-состояние Ребёнка (Ре) содержит все побуждения, которые возникают у ребёнка естественным образом. Оно также содержит запись ранних детских переживаний, реакций и позиций в отношении себя и других. Оно выражается как «старое» (архаическое) поведение детства. Эго-состояние Ребёнка отвечает также за творческие проявления личности.

Транзакции (информатика)

Транзакция (англ. *transaction*) — группа последовательных операций с [базой данных](#), которая представляет собой логическую единицу работы с данными.

Транзакция может быть выполнена либо целиком и успешно, соблюдая целостность данных и независимо от параллельно идущих других транзакций, либо не выполнена вообще, и тогда она не должна произвести никакого эффекта.

Транзакции обрабатываются [транзакционными системами](#), в процессе работы которых создаётся [история транзакций](#).

Различают последовательные (обычные), [параллельные](#) и [распределённые транзакции](#).

Распределённые транзакции подразумевают использование более чем одной транзакционной системы и требуют намного более сложной логики (например, two-phase commit — [двухфазный протокол фиксации транзакции](#)).

Также в некоторых системах реализованы [автономные транзакции](#), или подтранзакции, которые являются автономной частью родительской транзакции.

Распределенные транзакции

Задача двух генералов — в вычислительной технике [мысленный эксперимент](#), призванный проиллюстрировать проблему синхронизации состояния двух систем по ненадёжному каналу связи.

Откат (англ. rollback)

Системы обработки транзакций обеспечивают целостность базы данных при помощи записи промежуточного состояния базы данных перед её изменением, а затем, используя эти записи, восстанавливают базу данных до известного состояния, если транзакция не может быть совершена.

Прогон (англ. rollforward)

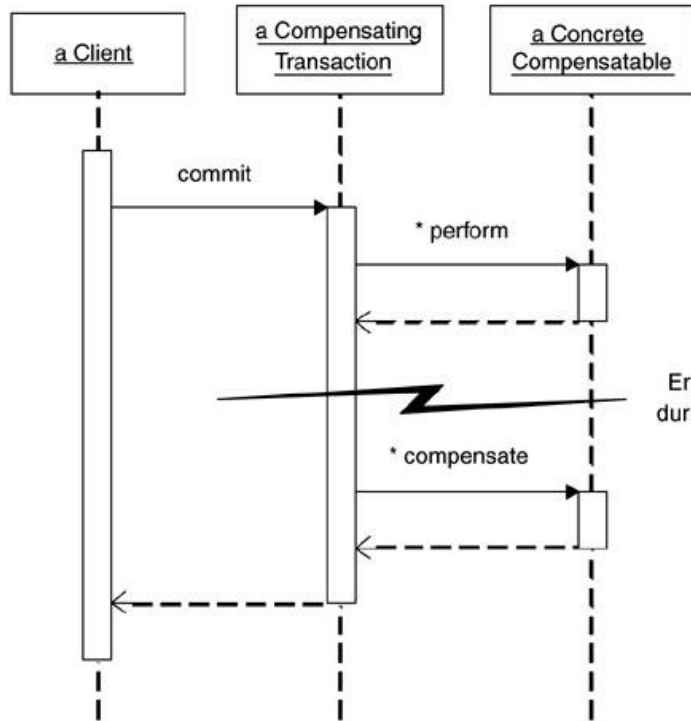
Кроме того, можно вести отдельный журнал всех изменений базы данных (иногда это называется after images); это не требует отката неудачных операций, но это полезно для обновления базы данных в случае отказа базы данных, поэтому некоторые системы обработки транзакций обеспечивают эту функцию.

Взаимная блокировка (англ. deadlocks)

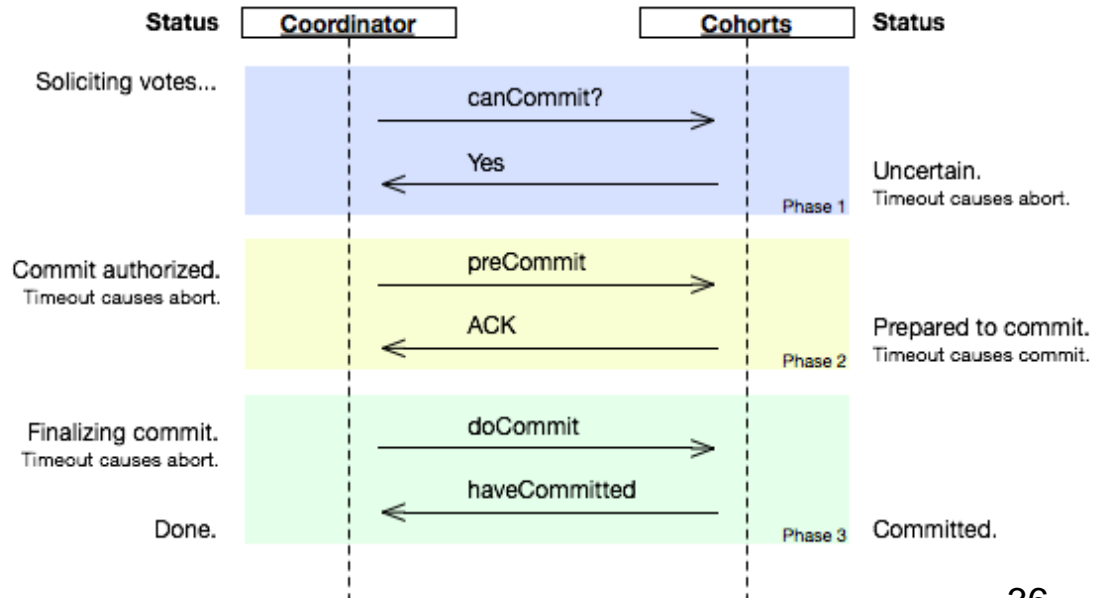
В некоторых случаях, две транзакции могут в ходе их обработки пытаться получить доступ к одной и той же части базы данных в одно и то же время, таким образом, что это будет препятствовать их совершению. Системы обработки транзакций предназначены для обнаружения таких ситуаций. Обычно обе транзакции отменяются и производится откат, а затем они автоматически запускаются в другом порядке, так что взаимоблокировка не повторится.

Транзакции

Отмена (abort) и компенсация (compensation)



Двухфазная обработка



Трёхфазная обработка

OLTP-системы

OLTP ([англ. Online Transaction Processing](#)), **транзакционная система** — обработка транзакций в реальном времени.

Способ организации базы данных, при котором система работает с небольшими по размерам транзакциями, но идущими большим потоком, и при этом клиенту требуется от системы минимальное время отклика.

Термин OLTP применяют также к информационным системам и приложениям. OLTP-системы предназначены для ввода, структурированного хранения и обработки информации (операций, документов) в режиме реального времени.

OLTP-приложениями охватывается широкий спектр задач во многих отраслях — [автоматизированные банковские системы](#), [ERP-системы \(системы планирования ресурсов предприятия\)](#), банковские и биржевые операции, в промышленности — регистрация прохождения детали на [конвейере](#), фиксация в статистике посещений очередного посетителя веб-сайта, автоматизация бухгалтерского, складского учёта и учёта документов и т. п. Приложения OLTP, как правило, автоматизируют структурированные, повторяющиеся задачи обработки данных, такие как ввод заказов и банковские транзакции. OLTP-системы проектируются, настраиваются и оптимизируются для выполнения максимального количества транзакций за короткие промежутки времени. Как правило, большой гибкости здесь не требуется, и чаще всего используется фиксированный набор надёжных и безопасных методов ввода, модификации, удаления данных и выпуска оперативной отчётности. Показателем эффективности является количество транзакций, выполняемых за секунду. Обычно аналитические возможности OLTP-систем сильно ограничены (либо вообще отсутствуют).

OLAP-системы

OLAP ([англ. online analytical processing](#), аналитическая обработка в реальном времени) — технология обработки данных, заключающаяся в подготовке суммарной (агрегированной) информации на основе больших массивов данных, структурированных по многомерному принципу.

Реализации технологии OLAP являются компонентами программных решений класса [Business Intelligence](#).

Многомерное моделирование является методом моделирования и визуализации данных как множества числовых или лингвистических показателей или параметров (measures), которые описывают общие аспекты деятельности организации.

Метод многомерного моделирования базируется на следующих основных понятиях:

Факт (Fact) — набор связанных элементов данных, содержащих метрики и описательные данные.

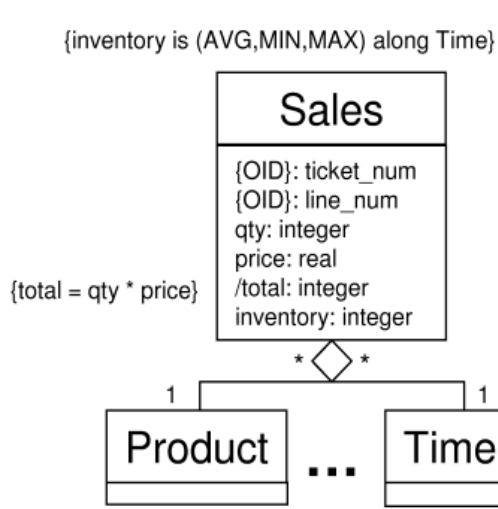
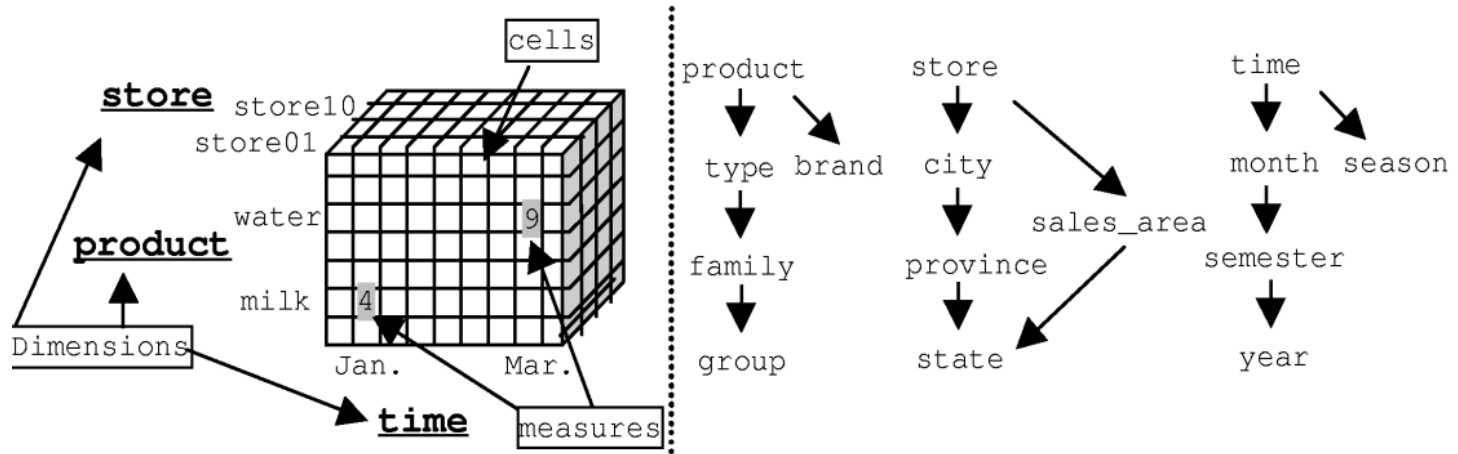
Атрибут (Attribute) – описание характеристики реального объекта предметной области.

Измерение (Dimension) — интерпретация факта с некоторой точки зрения в реальном мире.

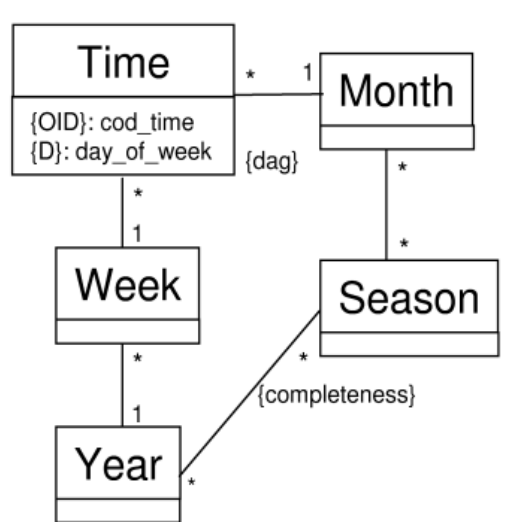
Параметр, метрика или показатель (Measure) — числовая характеристика факта

Гранулированность (Granularity) – уровень детализации данных.

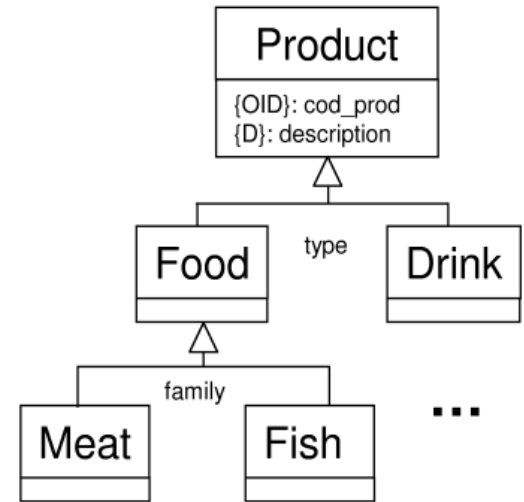
OLAP-системы



(a)



(b)



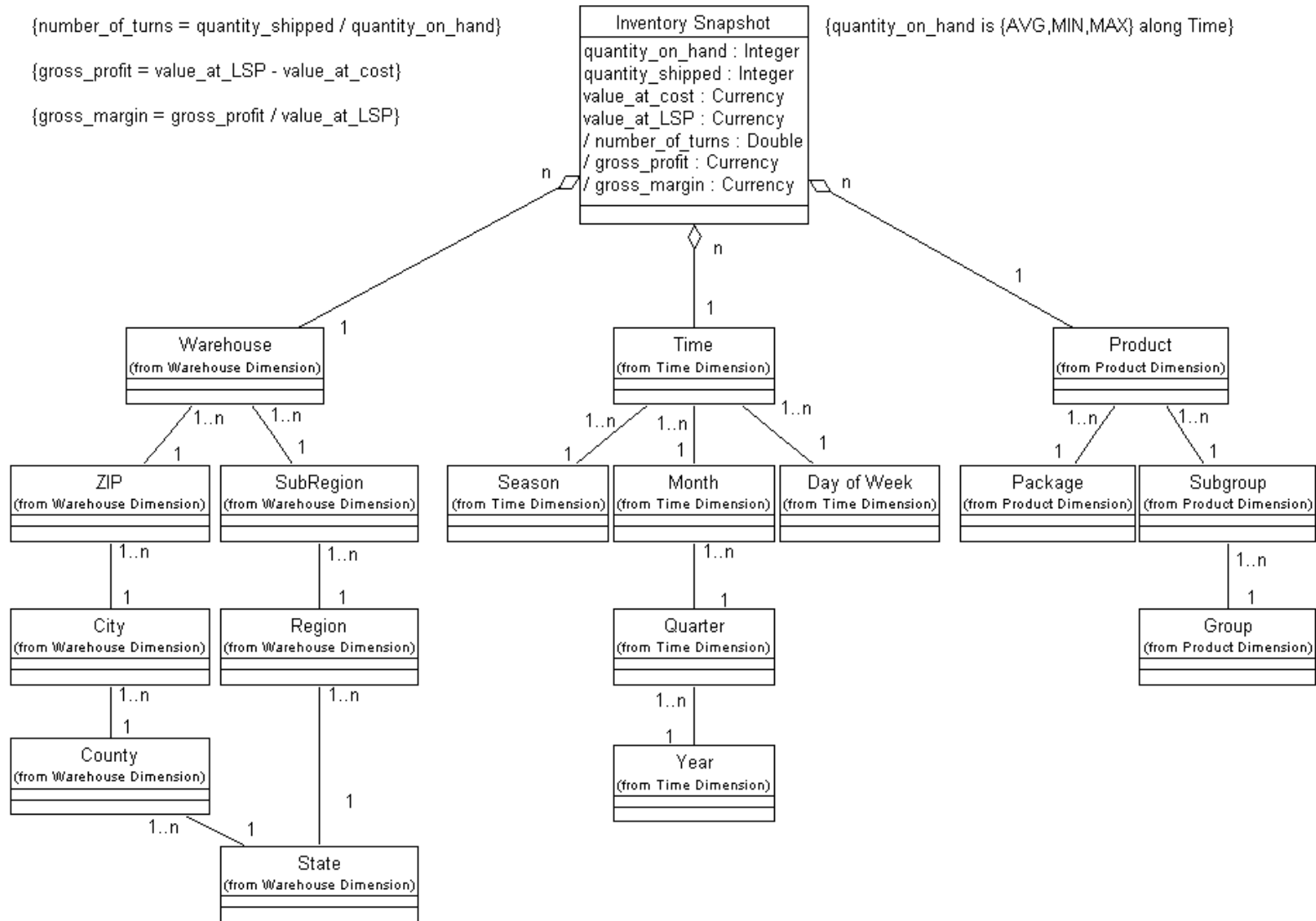
(c)

OLAP-системы

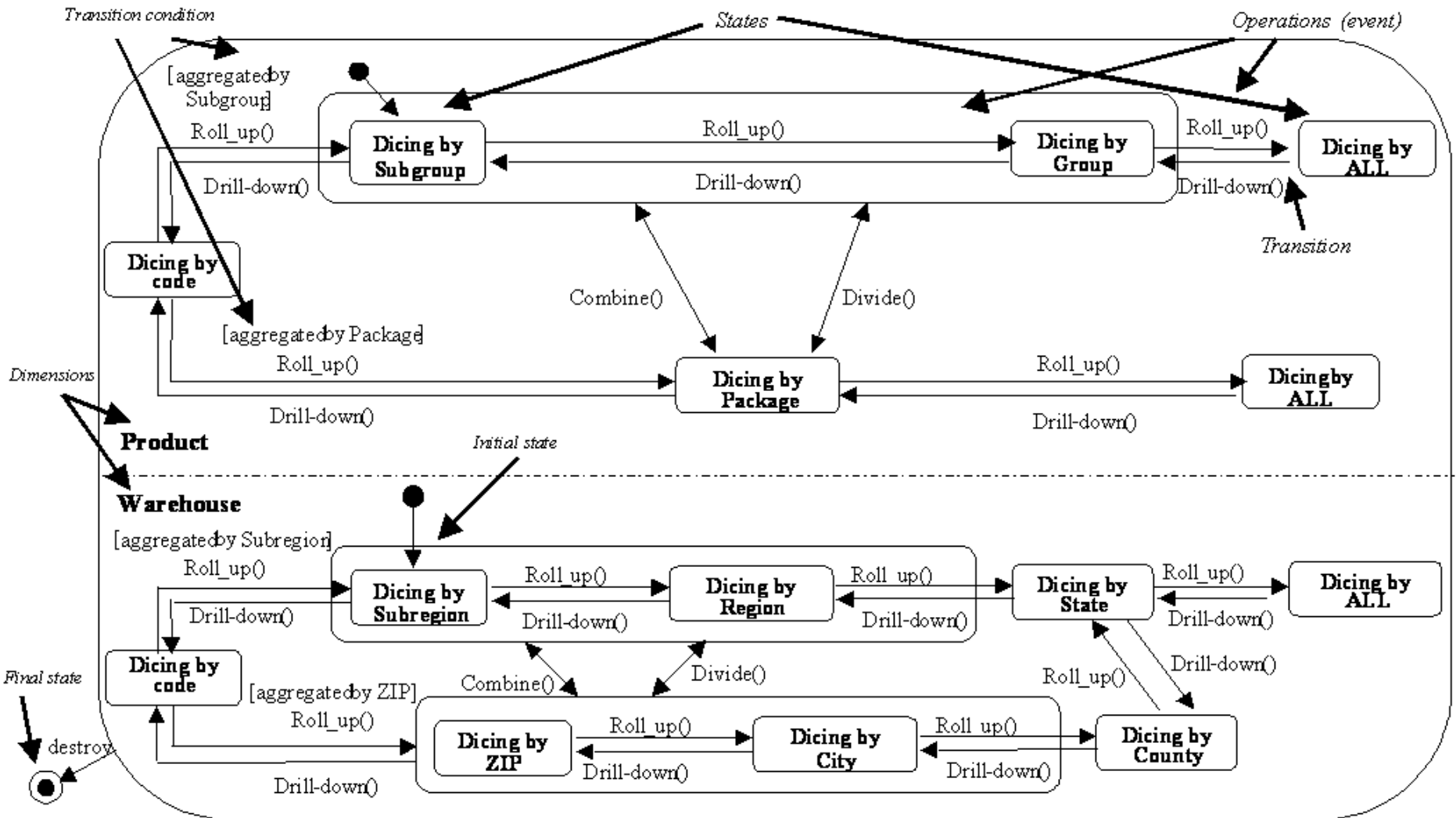
{number_of_turns = quantity_shipped / quantity_on_hand}

{gross_profit = value_at_LSP - value_at_cost}

{gross_margin = gross_profit / value_at_LSP}



OLAP-системы



Data Vault

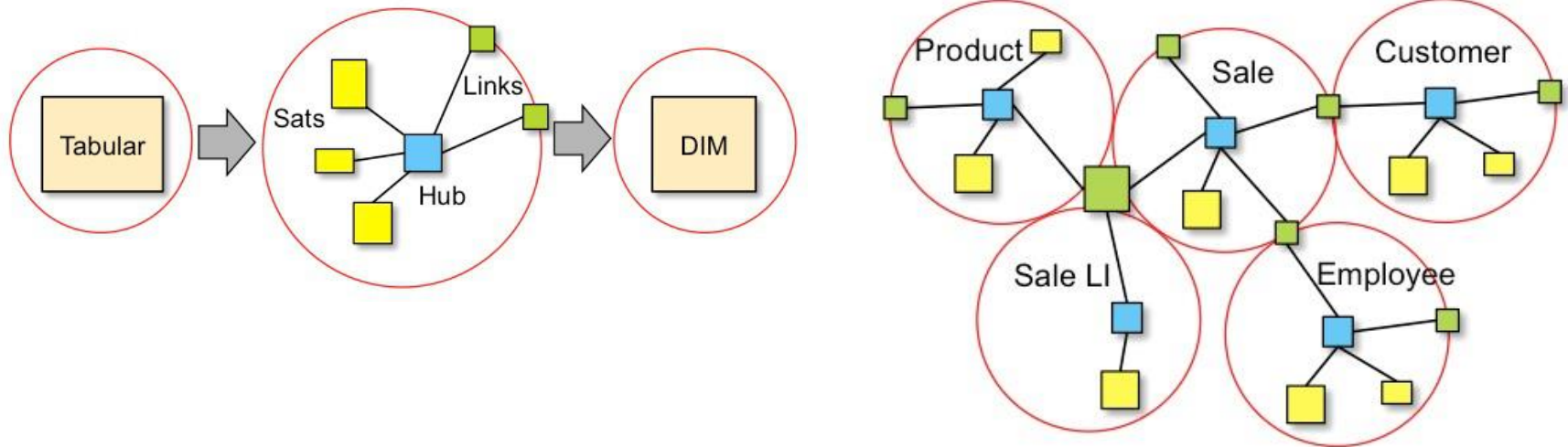
Модель Data Vault представляет собой набор связанных между собой нормализованных таблиц, ориентированных на хранение детализированной информации с возможностью отслеживания происхождения данных и поддерживающих одну или несколько областей бизнеса.

В модели Data Vault используется всего три типа таблиц:

Hub обеспечивает представление функциональных областей предметной области

Link обеспечивает транзакционную связь между Hub-таблицами

Satellite предоставляет детализацию первичного ключа Hub-таблицы



Системы классификации

Таксономия (от др.-греч. τάξις — строй, порядок и νόμος — закон) — учение о принципах и практике классификации и систематизации

Математически таксономией является древообразная структура классификаций определенного набора объектов. Вверху этой структуры — объединяющая единая классификация — корневой таксон — которая относится ко всем объектам данной таксономии. Таксоны, находящиеся ниже корневого, являются более специфическими классификациями, которые относятся к поднаборам общего набора классифицируемых объектов

Система классификации Линнея

Все живые существа можно классифицировать с помощью иерархической системы, в основе которой лежат категории рода и вида.

В системе Линнея все относится:

- либо к животным,
- либо к растениям,
- либо к неживой природе (минералам).

Таксономия системы:

Царство -> Тип -> Класс -> Отряд -> Семейство -> Род -> Вид



Системы кодирования

Система кодирования - строго определенный порядок присвоения условных обозначений единицам информации. Таким образом, все коды строятся по определенным правилам (системам).

Используемые для этих целей системы построения кодов подразделяются на:

- линейные (одномерные);
- шахматные.

Линейные системы кодирования

В линейных кодах условное обозначение соответствует только одной единице информации. По способу построения различают следующие линейные системы кодирования:

- порядковые,
- серийные,
- позиционные (разрядные, или десятичные),
- повторения,
- смешанные (комбинированные).

Шахматная система кодирования

В шахматных используются двухпозиционные коды, одновременно отражается характеристика двух информационных единиц (по строке и столбцу).

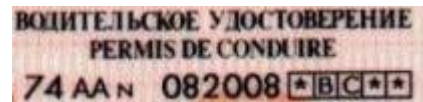
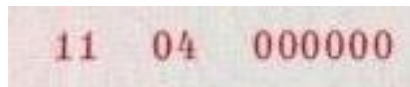
Системы кодирования

Порядковая система кодирования предполагает последовательное присвоение условных обозначений кодируемым единицам информации.

Специальной классификации информации, как правило, не требуется. Последовательность кодов задается, прежде всего, хронологией возникновения информационных единиц, но чаще всего объектом кодирования выступает информация, упорядоченная (систематизированная) по алфавиту.

Серийная система кодирования ориентирована на разделение классифицируемого множества по какому-либо признаку на отдельные части (серии). За каждой серией закрепляется своя группа условных обозначений (чисел, называемых номерами).

При этом номера единиц информации последующих серий не продолжают последовательно номера уже имеющихся единиц предыдущей серии, в результате создается определенный разрыв номеров, используемый в качестве резерва для последующего расширения (в случае необходимости) множества кодируемых позиций в каждой серии без нарушения общей логики построения списка.

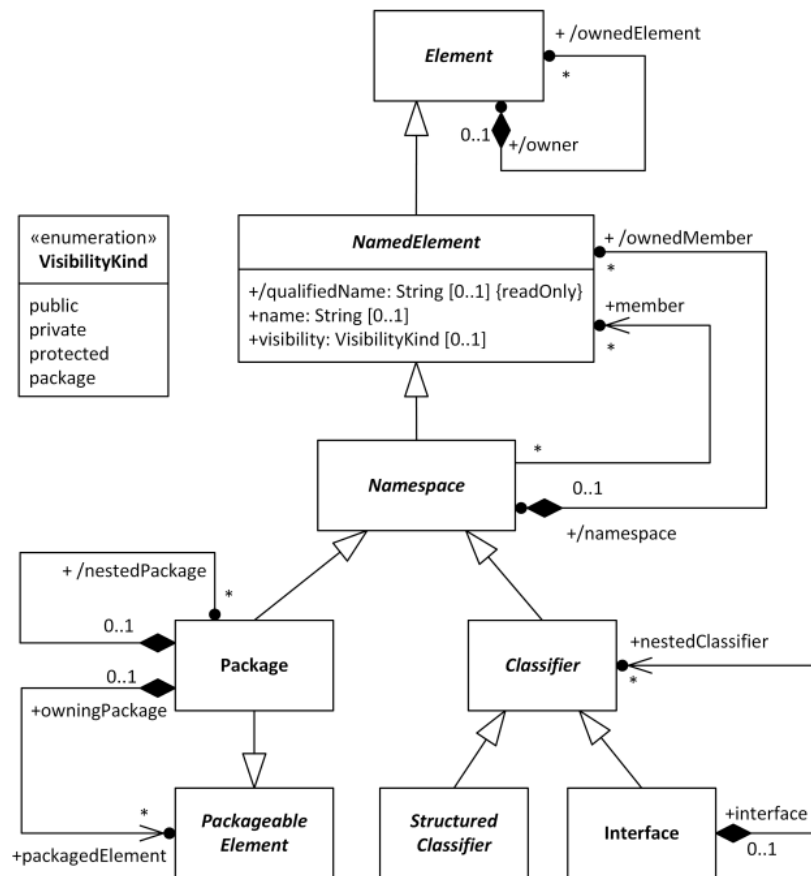
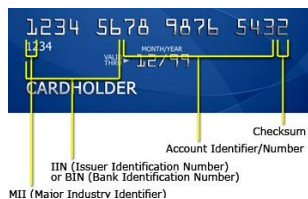


Системы кодирования

Позиционная (разрядная, десятичная) система кодирования предполагает иерархическую структуру представления информации или разделение ее по нескольким соподчиненным признакам.

Сущность данной системы заключается в том, что каждый уровень (или признак) классификации обеспечивается своей нумерацией в пределах всего уровня или признака (группы информации). При этом устанавливается предел разрядности группы и выбирается ее некоторая кратность.

Позиционная система применяется для кодирования сложных составных (иерархических) номенклатур, в которых, как правило, каждый реквизит, характеризующий низший уровень классификации, получает ряд характеристик, отражающих его принадлежность к более высокому уровню классификации.



Переходные ключи

Гармонизация общероссийского классификатора: Приведение общероссийского классификатора в соответствие с международной (региональной) классификацией, межгосударственным классификатором или международным (региональным) стандартом по классификации установленными путями гармонизации.

Переходной ключ: Таблица, устанавливающая соответствие каждой группировки или объекта классификации общероссийского классификатора одной или нескольким группировкам или объектам сопоставляемой классификации.

Код		Дробная единица ^[с 1] разряды	Наименования валют ^[с 2]		Наименования государств и территорий ^[с 3]
буквенный	цифровой		ОКВ	ISO 4217	в соответствии с ISO 4217 и/или ОКВ
Денежные единицы, включённые в действующую редакцию ОКВ					
AED	784	2	Дирхам (ОАЭ)	UAE Dirham	 ОАЭ
AFN	971	2	Афгани	Afghani	 Афганистан
ALL	008	2	Лек	Lek	 Албания
AMD ^[a 1]	051	2	Армянский драм	Armenian Dram	 Армения
ANG	532	2	Нидерландский антильский гульден	Netherlands Antillean Guilder	 Кюрасао  Синт-Мартен
AOA	973	2	Кванза	Kwanza	 Ангола
ARS	032	2	Аргентинское песо	Argentine Peso	 Аргентина

НСИ, MDM

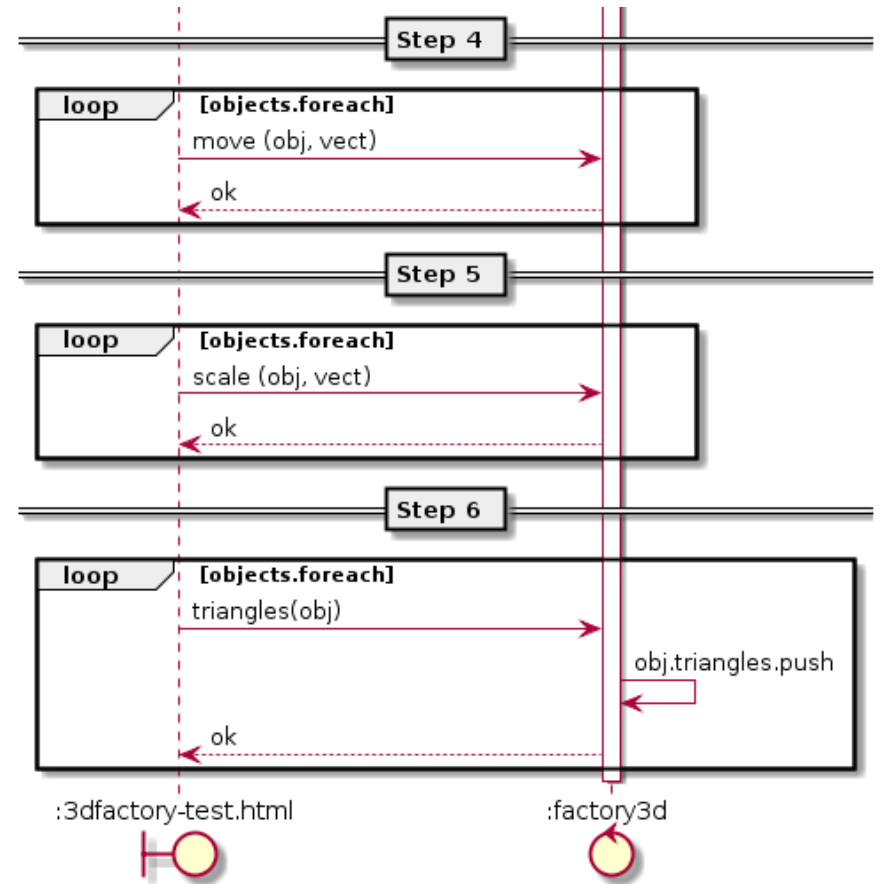
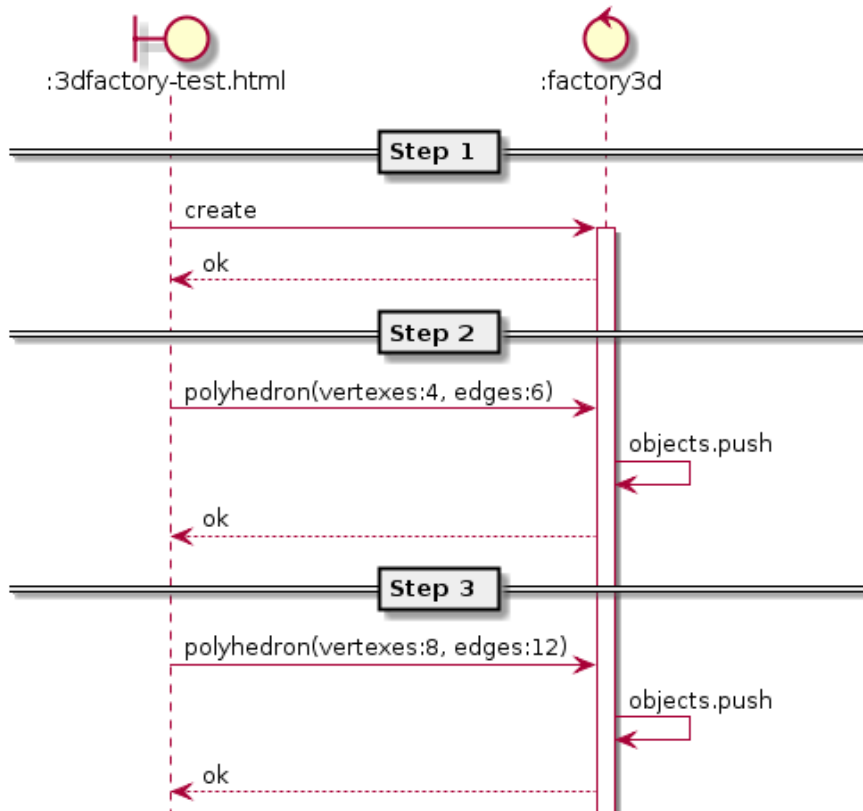
ГОСТ 34.003-90 Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения

Нормативно-справочная информация (НСИ) автоматизированной системы - Информация, заимствованная из нормативных документов и справочников и используемая при функционировании

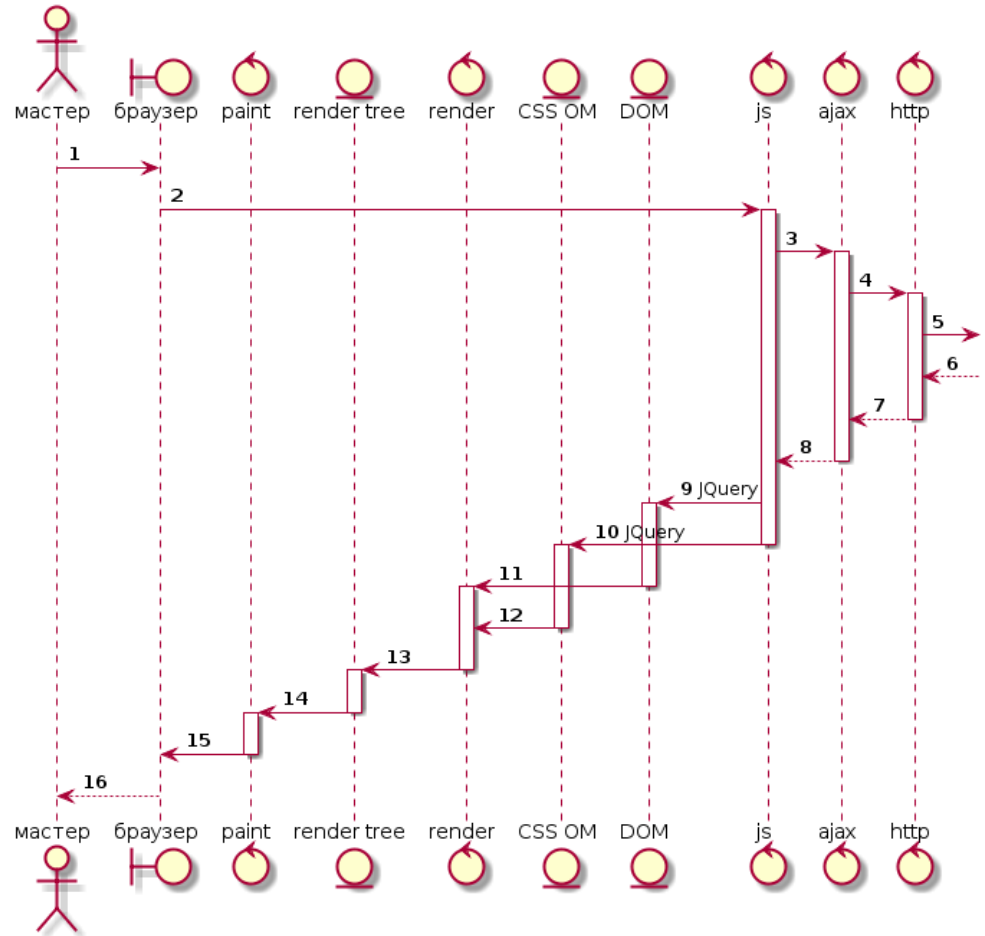
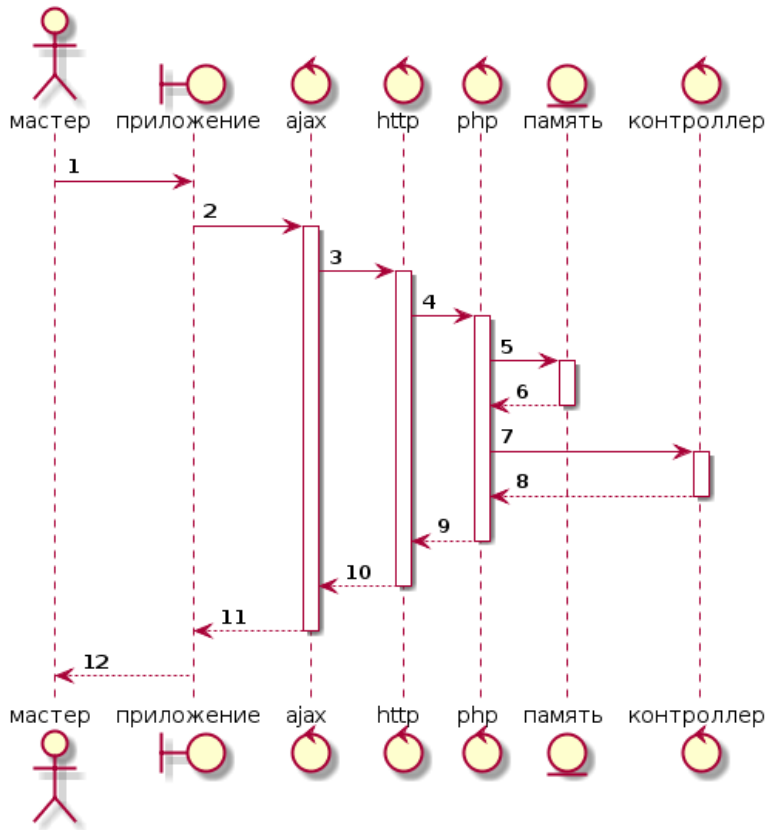
Управление основными данными, управление мастер-данными (англ. *Master Data Management*, MDM) — совокупность процессов и инструментов для постоянного определения и управления основными данными компании (в том числе справочными).

Мастер-данные — это данные с важнейшей для ведения бизнеса информацией: о клиентах, продуктах, услугах, персонале, технологиях, материалах и так далее. Они относительно редко изменяются и не являются транзакционными.

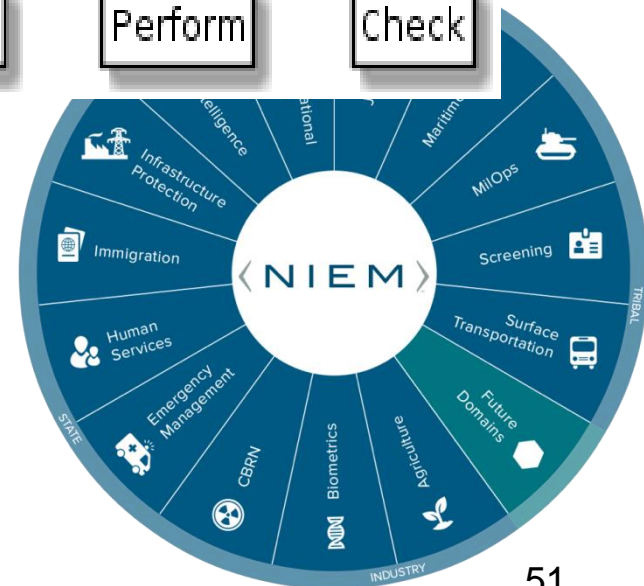
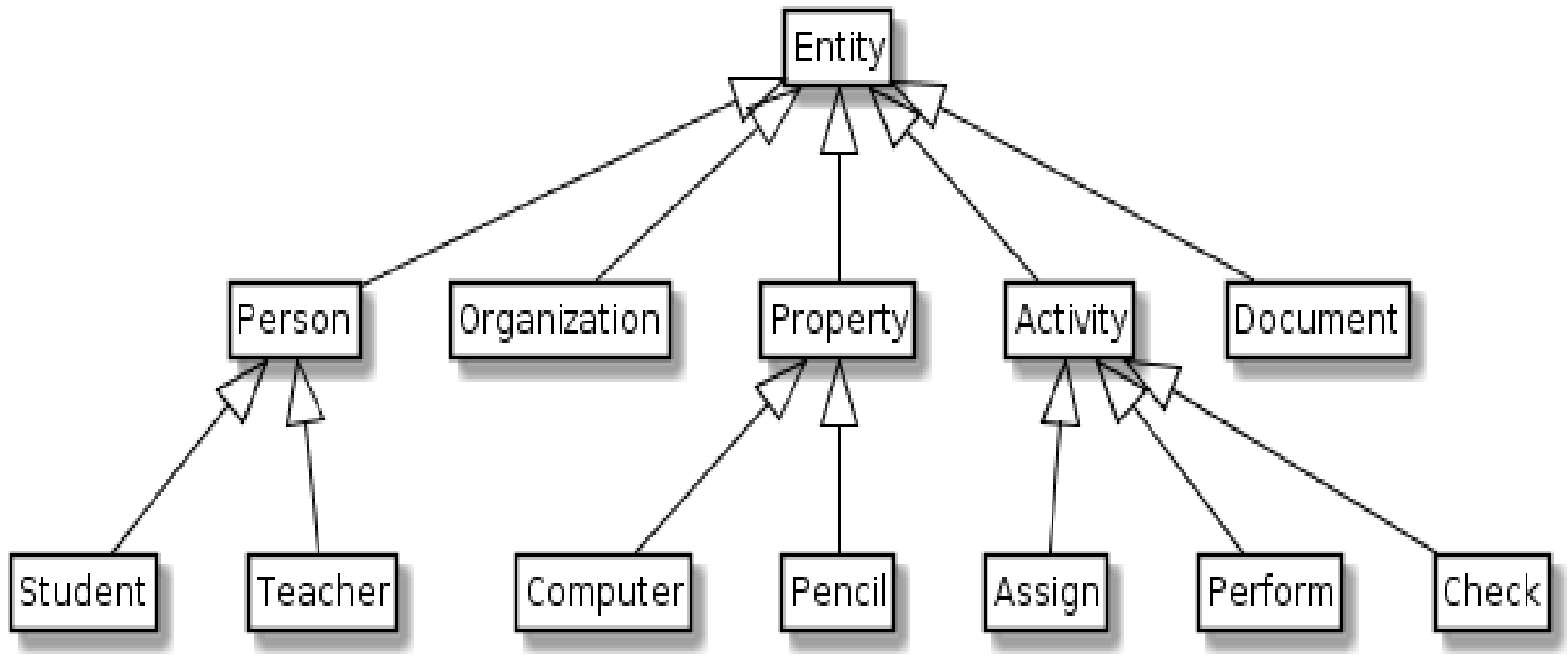
Проектирование взаимодействия



Проектирование взаимодействия



Проектирование взаимодействия



Проектирование взаимодействия

The Global JXDM endeavor began in March 2001 as a reconciliation of data definitions and evolved into a broad two-year effort to develop an XML-based framework that would enable the entire justice and public safety communities to effectively share information at all levels--laying the foundation for local, state, tribal, and national justice interoperability.

Approximately **16,000 justice and public safety-related data elements** were collected from various local and state government sources. These were analyzed and **reduced to around 2,000 unique data elements** that were then incorporated into about **300 data objects**, or reusable components, resulting in the Global Justice XML Data Dictionary (Global JXDD).

The Global JXDD components have inherent qualities enabling access from multiple sources and reuse in multiple applications. In addition, the standardization of the core components resulted in significant potential for increased interoperability among and between justice and public safety information systems.