

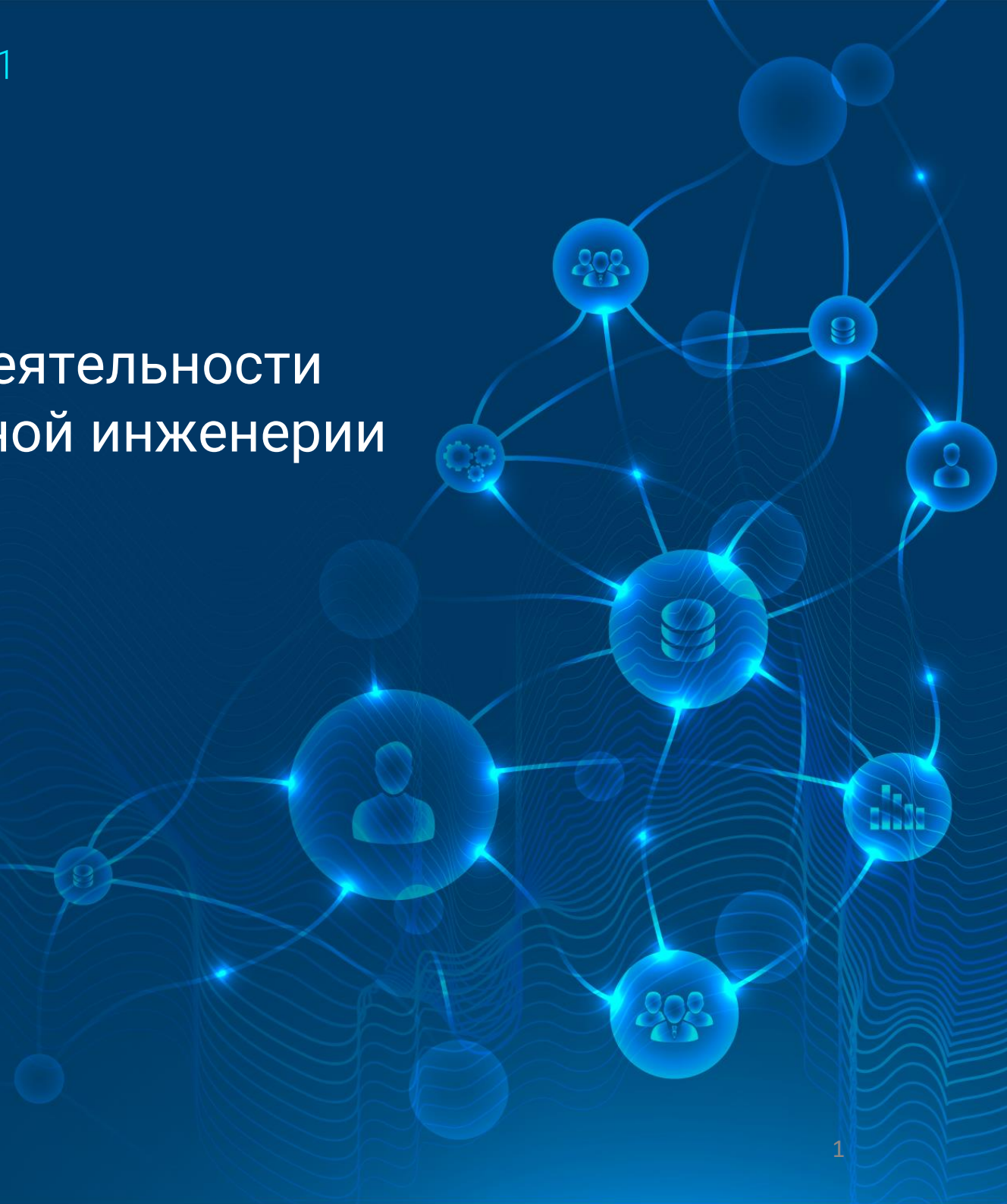
Построение верхнеуровневой модели деятельности компании на основе принципов системной инженерии

Спикер:

Дмитрий Пинаев

Генеральный директор

Современные технологии управления



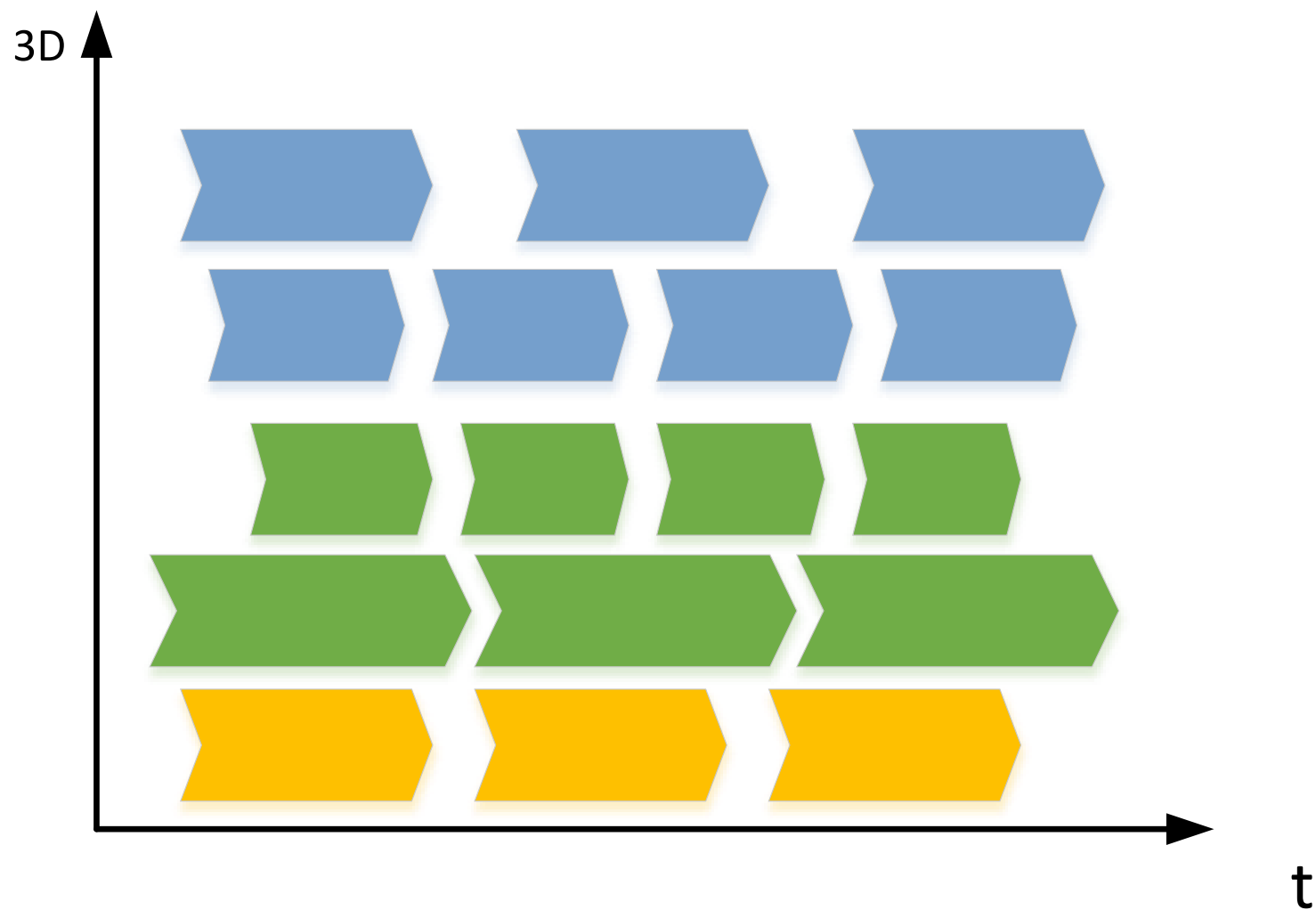


В качестве основных теорий для рассуждения и обоснования предлагаемых решений используются:

- Понятие «Онтология» и теория множеств для конструирования классов онтологии.
- 4D-экстенционализм.
- Нотация для экземпляров и классов (Matthew West «Developing high quality data models», ISO 15926-2 «Industrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities — Part 2: Data model»)
- Понятия и методы системной инженерии.



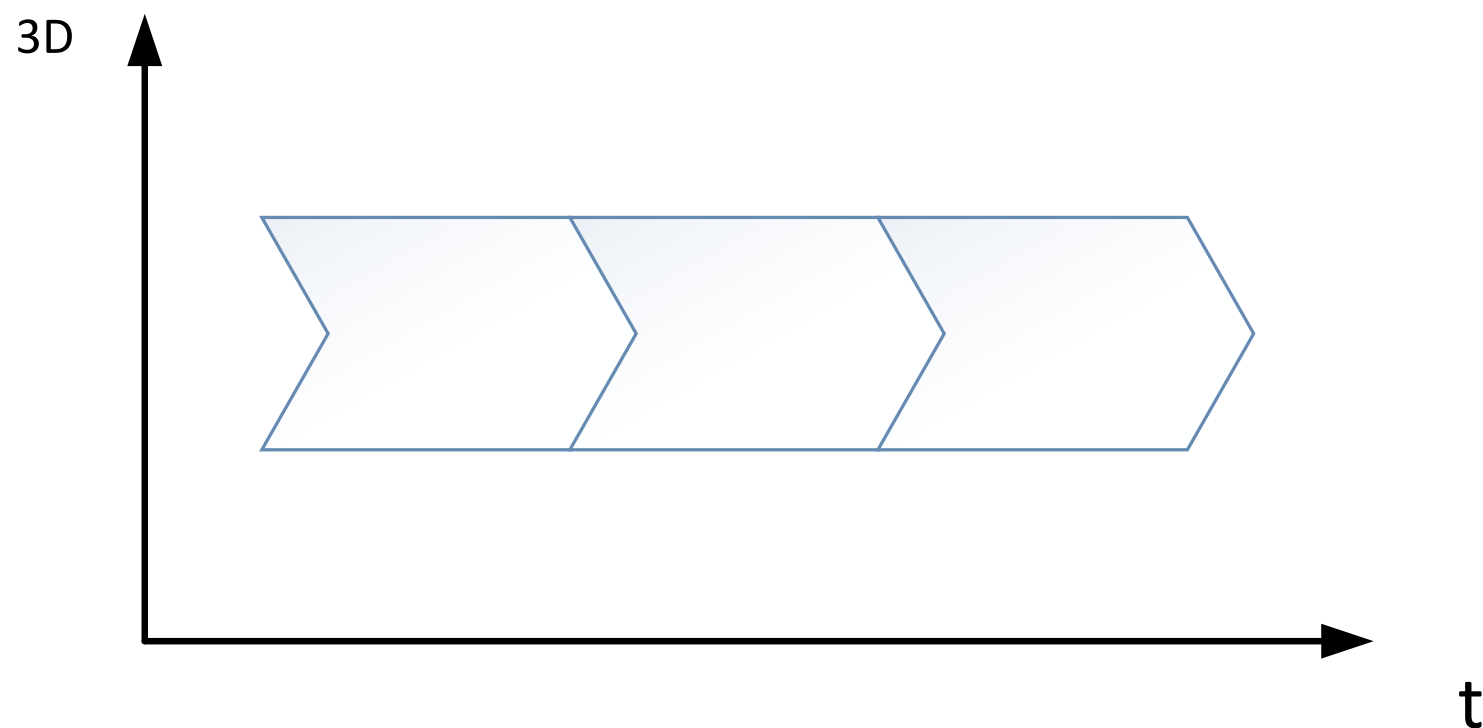
Постановка задачи



Ежедневно в компании совершаются тысячи, а то и миллионы конкретных (имеющих время начало и конца, а также место выполнения) операций.



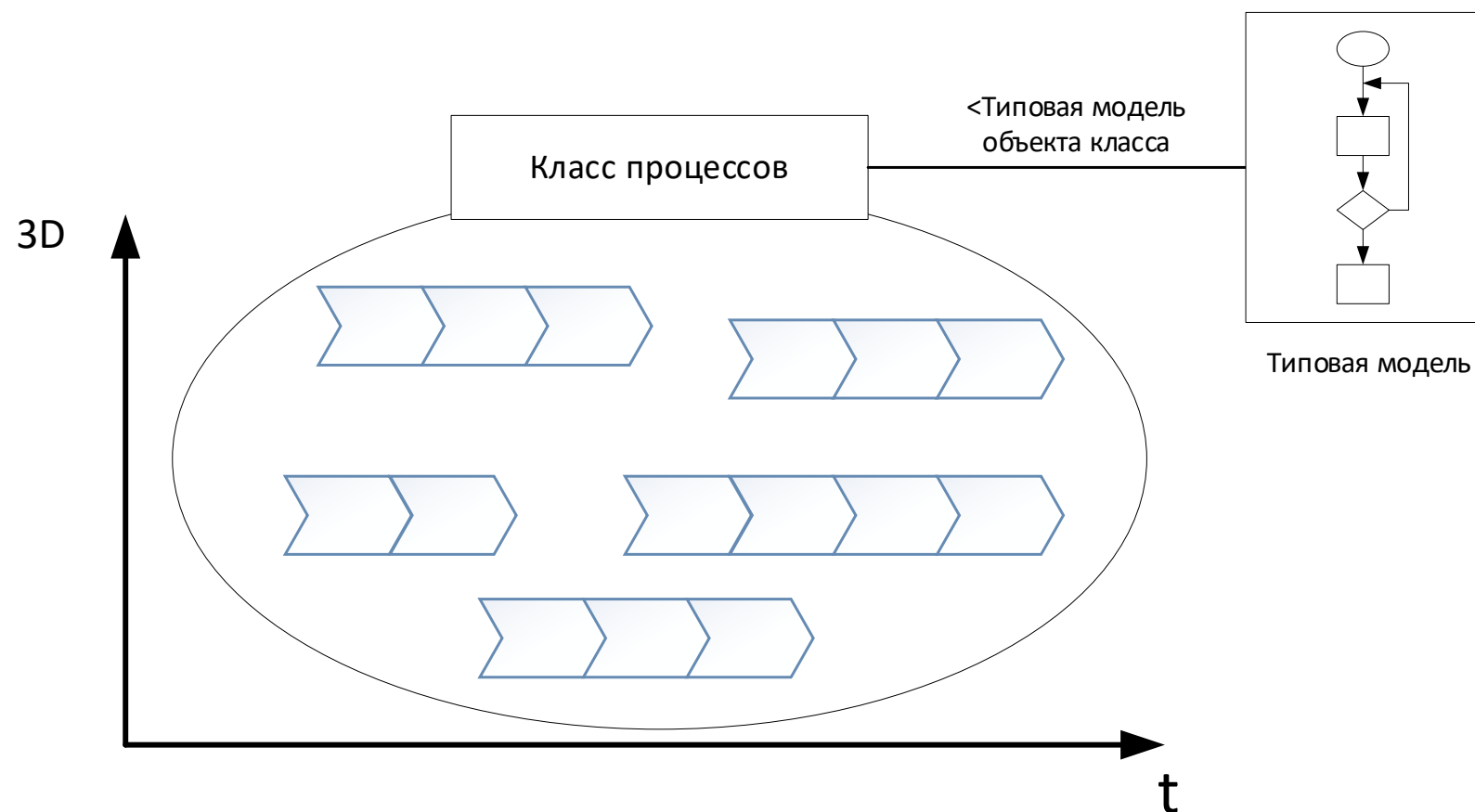
Постановка задачи



Конкретный процесс тоже является операцией и также имеет время и место выполнения. Просто процесс – это такая операция, структура которой известна субъекту, или он предполагает существование её структуры.



Постановка задачи



Процессы могут происходить в разное время и в разном месте. В класс мы объединяем те, которые считаем похожими. И при этом количество операций в их составе может быть разным! Вспомните: на диаграмме процесса (типовой модели) есть развилки и циклы, которые приводят к разному составу операций у конкретных процессов в реальности.



Требования к модели верхнего уровня (bird's-eye view) :

1. Модель может поместиться в голову человека «целиком».
2. Модель должна быть понятна неподготовленному человеку. (А, в идеале, подходить на роль учебного материала для новичков.)
3. Модель должна давать четкий ответ на вопрос: что именно делает компания. А для этого очень желательно, чтобы выполнялось требование 4.
4. Модель должна моделировать с

Наша неспособность дать простое описание, а следовательно, и обеспечить понимание таких систем (прим. автора: систем средней сложности) делает их проектирование и создание трудоемким и дорогостоящим процессом и повышает степень их ненадежности. С ростом технического прогресса адекватное описание систем становится все более актуальной проблемой.

*Дэвид А. Марка и Клемент МакГоуэн
Методология структурного анализа и проектирования (SADT)*



Пример фреймворка для верхнеуровневой модели

APQC Process Classification Framework (PCF) - Cross Industry - Version 7.2.1

September 26, 2019

- Создавалась для бенчмаркинга компаний (сравнения между собой)
- 13 верхнеуровневых категорий процессов: 6 операционных и 7 управления и поддержки
- Глубина: местами до 5 уровня. Всего 1264 элемента (Activity) на 4 уровне.

Критика

- Переусложненная онтология: 2 сущности для группировки процессов (Category, Process Group, 3 сущности для декомпозиции процессов (Process, Activity, Task).
- Понятность модели вызывает вопросы...
- Строго заданное фиксированное количество уровней.
- Нет связи по входам и выходам!!!

PCF LEVELS EXPLAINED

Level 1 - Category

1.0 Develop Vision and Strategy (10002)

Represents the highest level of process in the enterprise.

Level 2 - Process Group

1.1 Define the business concept and long-term vision (17040)

Indicates the next level of processes and represents a group of processes.

Level 3 - Process

1.1.5 Conduct organization restructuring opportunities (16792)

A process is the next level of the decomposition after a process group. This can include core elements needed to accomplish the process as well as element related to variants and rework.

Level 4 - Activity

1.1.5.3 Analyze deal options (16795)

Indicates key events performed when executing a process.

Level 5 - Task

1.1.5.3.1 Evaluate acquisition options (16796)

Tasks represent the next level of hierarchical decomposition after activities. Tasks are more fine grained and vary widely across industries.

OPERATING PROCESSES

1.0	Develop Vision and Strategy
1.2	Develop business strategy
1.2.5	Create organizational design
1.2.5.1	Evaluate breadth and depth of organizational structure
1.2.5.2	Perform job-specific roles mapping and value-added analyses
1.2.5.3	Develop role activity diagrams to assess hand-off activity
1.2.5.4	Perform organization redesign workshops
1.2.5.5	Design the relationships between organizational units
1.2.5.6	Develop role analysis and activity diagrams for key processes
1.2.5.7	Assess organizational implication of feasible alternatives
1.2.5.8	Migrate to new organization

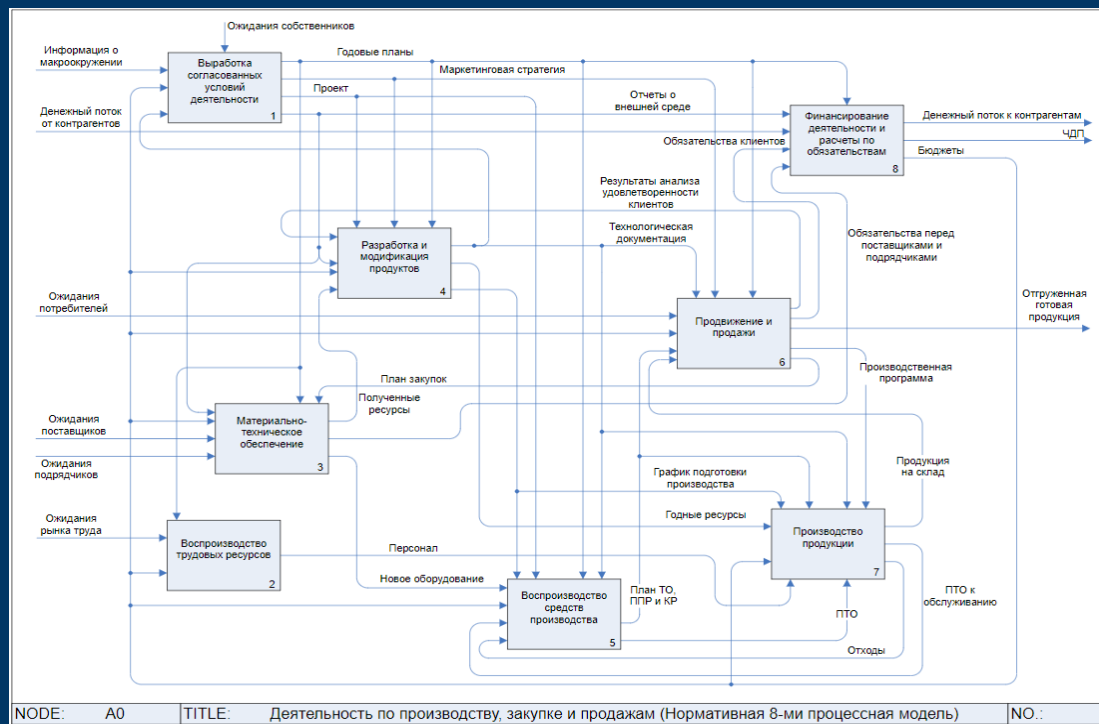


Типовые модели экосистемы Business Studio

- Типовые структуры процессов СТУ

- Оказание услуг (PDF)
- Проектная деятельность (PDF)
- Производство (PDF)
- Управляющая компания (PDF)

- 8-процессная модель Тимура Кадыева



- Ландшафт процессов





Процессы или функции?

Одна организация, называемая APQC, опубликовала межотраслевую рамочную модель классификации процессов (Process Classification Framework, далее PCF), – иерархическую структуру, состоящую из категорий, групп процессов, отдельных процессов и видов деятельности. К сожалению, очень немногие из процессов и видов деятельности, перечисленные в PCF соответствуют понятиям процесс и деятельность, в том значении, котором их определяет BPMN. Большинство из них относятся к постоянно протекающим бизнес-функциям типа «управлять X», а не действиям в дискретных экземплярах процесса с четко определенным началом и окончанием...

...У меня нет цели придраться именно к APQC. Эта проблема широко распространена в литературе по бизнес-архитектуре и управлению бизнес-процессами. Мне приходилось сталкиваться с ситуациями, когда команда BPM-архитекторов определяла перечень основных "видов деятельности", которые не являлись дискретными повторяющимися действиями, с четко определенными начальными и конечными точками, а потом поручала специалистам по моделированию процессов связать их вместе, чтобы описать процесс от начала до конца, что не представляется возможным.

*Bruce Silver
Bpmn Method and Style*



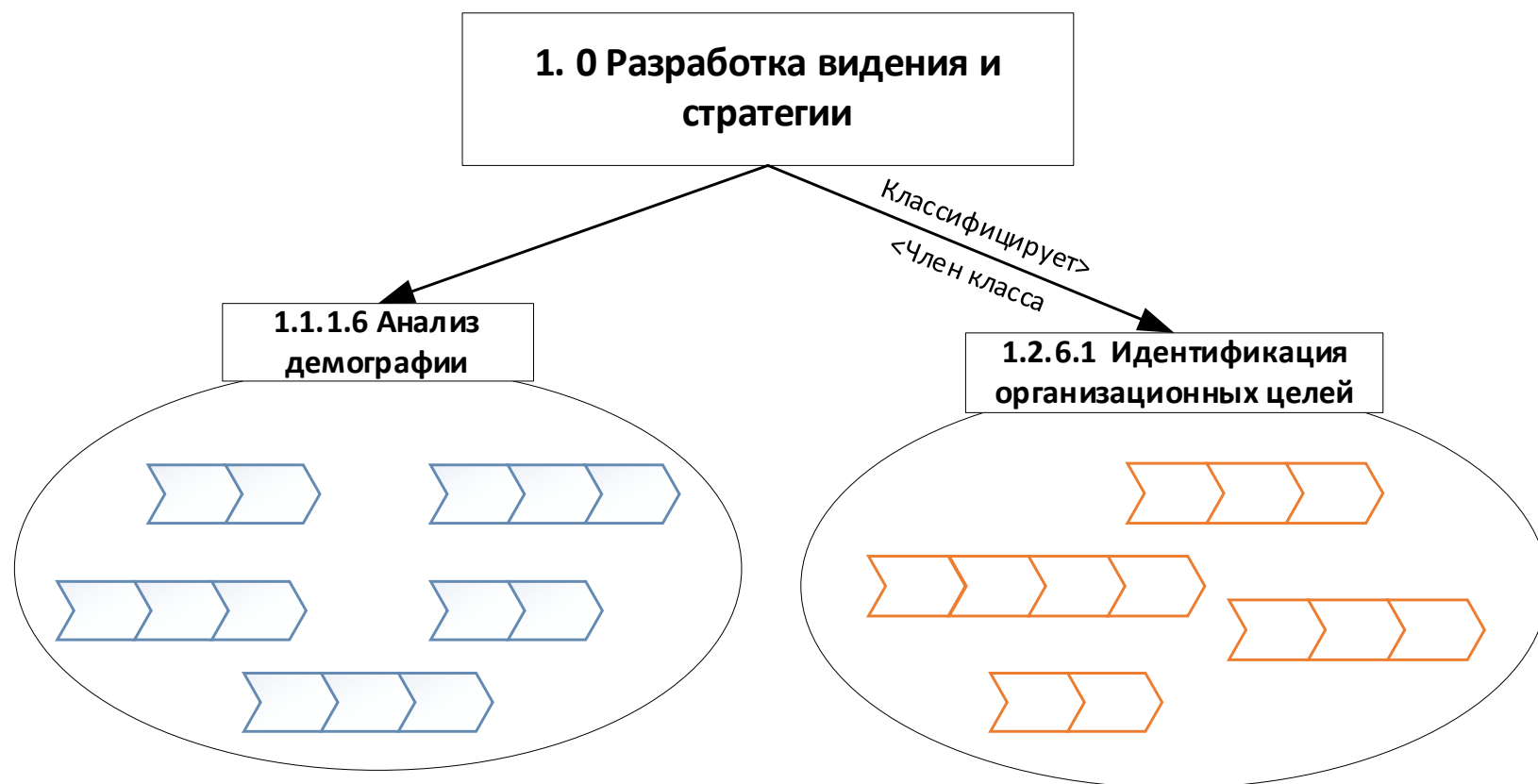
Процессы или функции?

Что мы делаем при объединении классов процессов в верхнеуровневой модели в нечто большее:

- Композиция?
- Агрегация?
- Классификация (включение объекта в класс)?



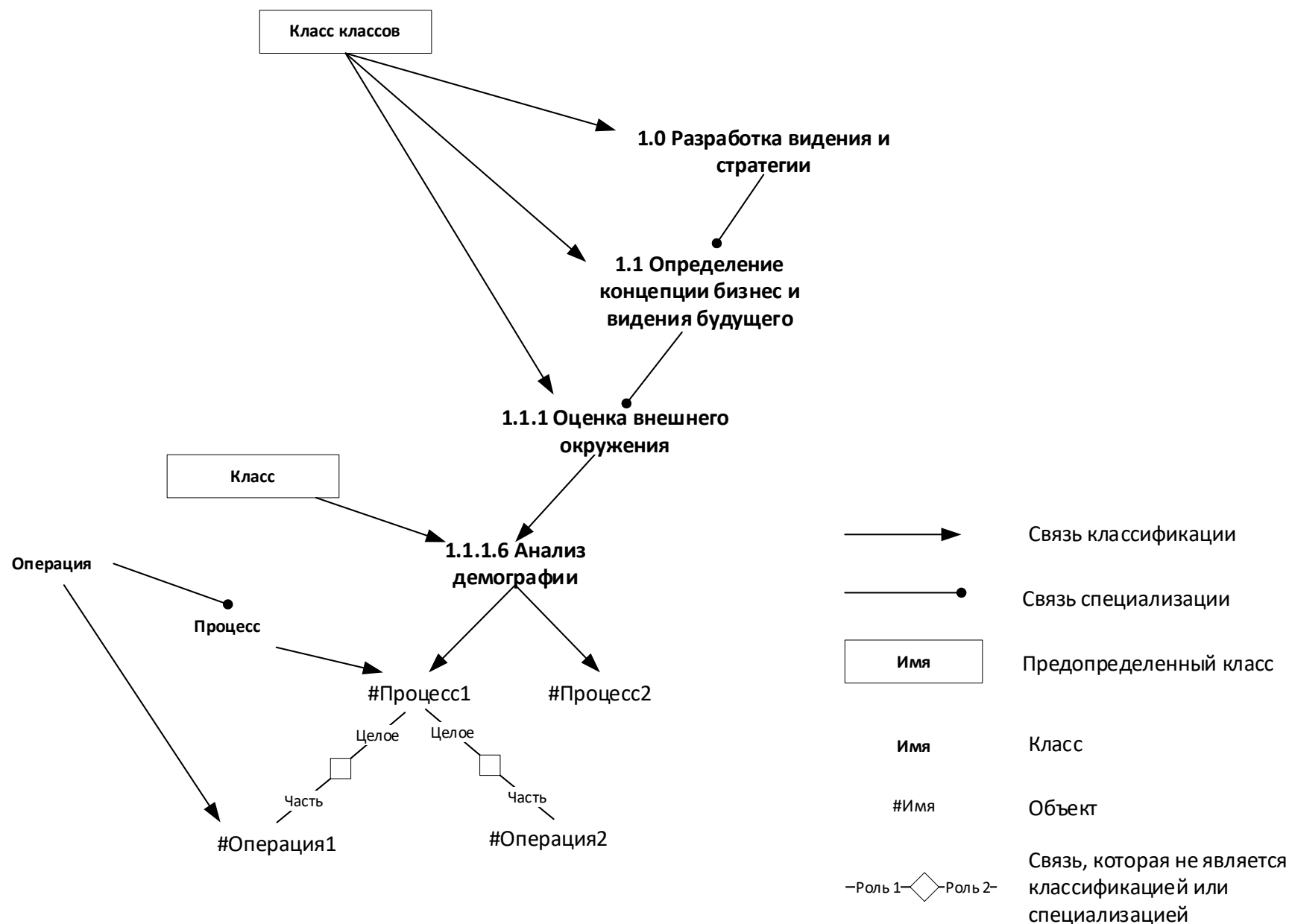
Процессы или функции?



- Под классом мы будем понимать множество его членов. Члены включаются в класс на основе набора критериев.
- Класс носит вневременной характер, он существует всегда. По классу нельзя «постучать пальцем», это не объект.
- В онтологии, базирующейся на теории множеств, возможно включение класса в качестве члена в другой класс. В данном примере в класс «Разработка видения и стратегии» мы включаем классы процессов именно в качестве членов.



Процессы или функции?



С использованием всего двух понятий «Операция» и «Процесс» была разработана простая онтология, которая позволяет осуществлять декомпозицию деятельности на неограниченную глубину.



Использование нотации IDEF0 для создания моделей деятельности верхнего уровня

Функциональная модель представляет с требуемой степенью детализации систему функций, которые в свою очередь отражают свои взаимоотношения через объекты системы.

Методология структурного анализа и проектирования (SADT)

Function: An activity, process, or transformation (modeled by an IDEF0 box) identified by a verb or verb phrase that describes what must be accomplished.

INTEGRATION DEFINITION FOR FUNCTION MODELING (IDEF0)

Arrows do not represent flow or sequence as in the traditional process flow model.

INTEGRATION DEFINITION FOR FUNCTION MODELING (IDEF0)

Вывод

Функция – это класс классов процессов.

(Или, с т.з. понятий ООП, если есть процесс и его экземпляры, то:

Функция - это класс процессов).



Использование нотации IDEF0 для создания моделей деятельности верхнего уровня

Следствия:

1. Функцию мы можем собрать из любых классов процессов, как нам заблагорассудится.
Мы теперь можем легко решить очередную философскую проблему - что такое бухгалтерский учет – процесс или не процесс? Бухгалтерский учет – это, конечно, не процесс. Это функция, включающая все классы процессов, относящихся к предметам бухгалтерского учета.
2. Один класс процессов может входить в несколько функций (и это не грех, как принято считать сейчас. Мир слишком многогранен, чтобы его можно было описать одной таксономией).
3. Мы можем обобщать функции до бесконечности вверх. На нижнем уровне, когда в функцию оказывается включен только один класс процессов, обычно осуществляют смену типа модели и переходят к созданию типовой модели процессов класса (описывающей все процессы класса). При этом дальнейшее разбиение на части происходит с использованием связи «композиция».
4. Стрелки между функциями моделируют объекты, перемещающиеся между функциями. Это значит, что в реальности найдутся два конкретных процесса между которыми будут перемещаться конкретные объекты.



Замечания об выработанной онтологии

Онтология не противоречит IDEF0, а также языкам BPMN и Archimate:

...процессом, в BPMN, называется последовательность действий от исходного состояния экземпляра процесса до некоторого определенного конечного состояния. Начало процесса отмечается инициирующим событием, таким, как, например, получение запроса. Модель процесса – это карта всех возможных маршрутов, или последовательностей действий: от исходного события до какого-либо определенного конечного состояния: успешного завершения или исключения. Как и деятельность, процесс дискретен, а не непрерывен. В ходе деловой деятельности он происходит неоднократно и имеет четко определенные начало и окончание. Каждый экземпляр процесса следует по некоторому маршруту в модели процесса от его начала до окончания.

*Bruce Silver
Bpmn Method and Style)*

Бизнес-процесс

Последовательность элементов поведения бизнес-слоя, обеспечивающая достижение определенного результата. Например, заданный набор продуктов или бизнес-сервисов.

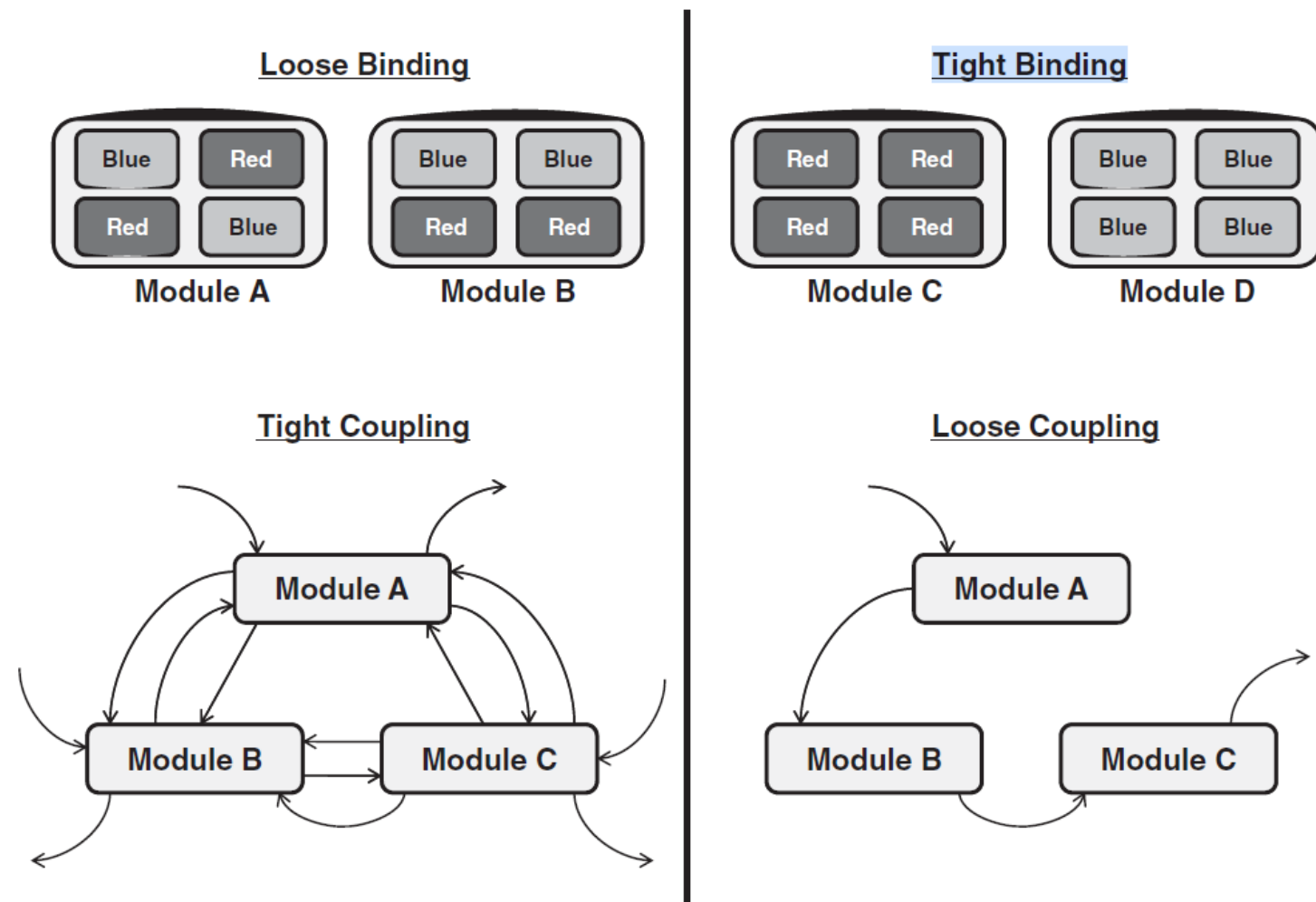
Бизнес-функция

Набор элементов поведения бизнес-слоя, выделенный на основе заданного критерия. Например: необходимые бизнес-ресурсы или компетенции. Структура бизнес-функций может соответствовать организационной структуре, но не обязательно явно повторяет организационную структуру.

Язык Archimate 3.1



Принцип «модульности» при проектировании архитектуры



Сильная сцепленность - слабая связанность» из программирования

При модульном разбиении системы «сильно сцепленные» функции группируются в слабо «связанные модули».

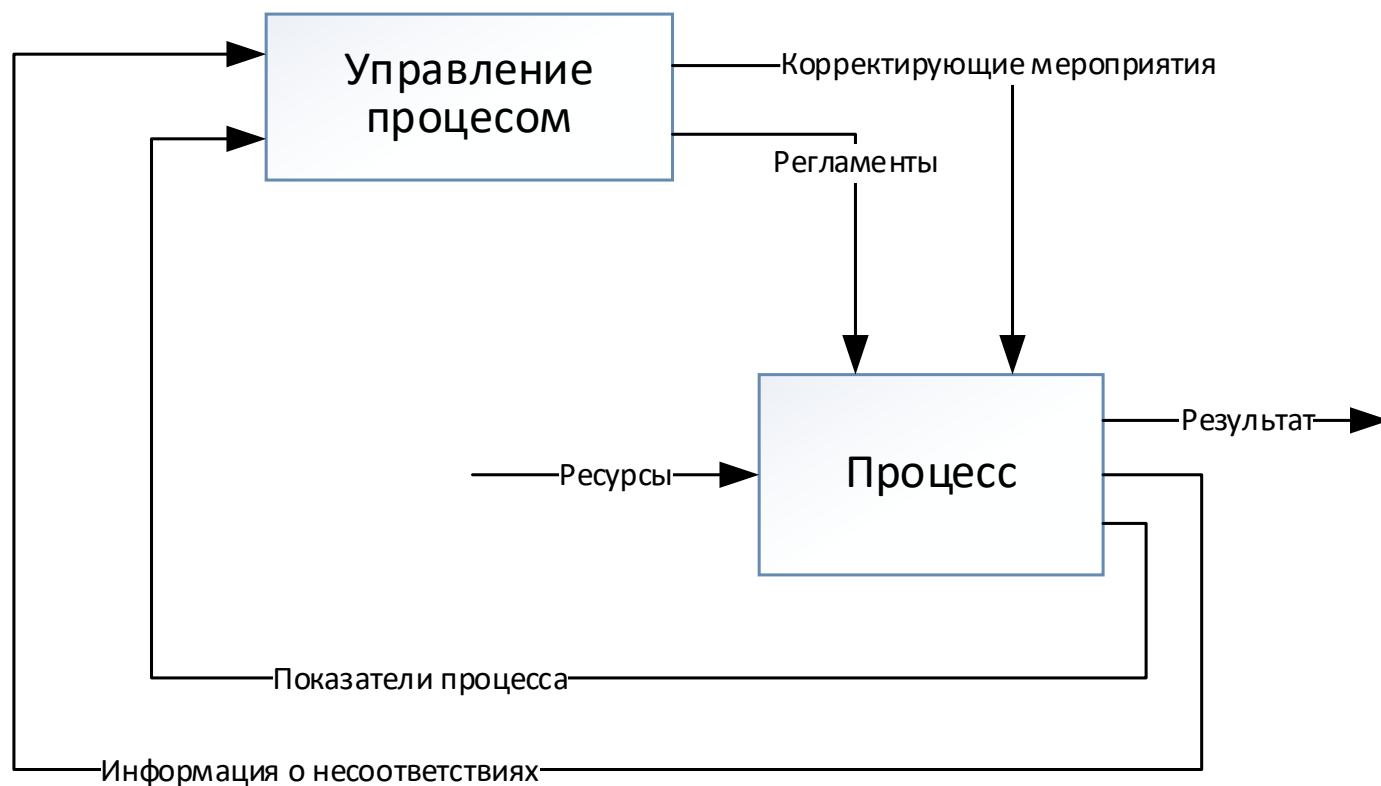
Александр Косяков
Системная инженерия. Принципы и практика.

Это обозначение (прим., "SA-блок»), четко выражало одну важную идею, связанную с тем, что сегодня называется иерархической многоуровневой модульной системой. Каждый уровень представлял собой законченную систему (блок), поддерживаемую и контролируемую системой (блоком), находящейся над ней...

Дуглас Т. Росс,
Методология структурного анализа и проектирования (SADT)



Принцип «автономности» при проектировании архитектуры организационной системы



Пример одного из существующих фреймворков

Насколько с точки зрения здравого смысла корректно смешивать и показывать в одной модели, как мы едем на автомобиле и одновременно ремонтируем его? Или как работает кофеварка и как она себя улучшает? В технических системах это кажется абсурдным. Но в моделях организации это является распространенным подходом.



Принцип «автономности» при проектировании архитектуры организационной системы

Система					
Замысел	Разработка	Производство	Функционирование	Обслуживание	Утилизация

Вариант жизненного цикла системы

(Источник: ISO 15288:2002 Life Cycle Management — System Life Cycle Processes, редакция именно 2002 года)





Принцип «автономности» при проектировании архитектуры организационной системы





Принцип «автономности» при проектировании архитектуры организационной системы



Принцип автономности заключается в том, что мы делим бизнес-систему на подсистемы, которые способны существовать друг без друга. Отношение между этими системами не горизонтальные по входам и выходам, как в технологической цепочке, а вертикальные:

- Отношение «Создание» – система физически создает другую систему.
- Отношение «Обслуживание» – система ремонтирует другую систему.



Другие принципы декомпозиции

Жизненный цикл – декомпозиция по стадиям ЖЦ ключевого объекта рассматриваемой деятельности. Например: потребитель должен пройти через стадии:

- потенциальный;
- информированный;
- привлеченный;
- обсуживаемый;
- удовлетворенный.

Технологическая цепочка – декомпозиция деятельности по технологическим этапам на основе логических и физических принципов («не сделав А, нельзя сделать Б»). Этапы связаны горизонтальными входами-выходами: следующему этапу нужны объекты, полученные на предыдущем этапе.

В качестве вспомогательного метода можно применять подход цепочки создания ценности. Но он не может использоваться как основной принцип, т.к. декомпозиция происходит с точки зрения потребителя, а не архитектора системы.



Другие принципы декомпозиции

- **Без синхронизации** – процесс запускается, если у него на входе появился результат или пришел сигнал (в любой форме) на старт работы из предыдущего процесса.
- **Синхронизация без планирования** – вытягивающая система на основе карточек «канбан». Известен в двух формах:
 - Процесс запускается, если его результат нужен в следующем процессе.
 - Процесс запускается (объект работы затягивается в систему из очереди), если в системе освободился ресурс. Эта вариация используется в софтовой разработке.
- **Синхронизация с планированием** – предварительный расчет планов на заданный горизонт планирования на основе информации о длительности работ. Используется, когда необходимо синхронизировать множество работ между собой и получить минимальное общее время и/или затраты. В качестве примера можно привести масштабное планирование производственной деятельности по методу MRP2. Функция планирования при этом явно помещается в модель.



Модификация нотации IDEF0

Класс стрелок «Управление»

1. Стрелки «Управление» в первую очередь используются для моделирования объектов, запускающих процесс (планы, задания) в том случае, когда для инициации не используется сам факт появления объекта на входе (см.п.2).
2. В случае наличия единственного трансформируемого объекта на входе, я не согласен с требованием стандарта указывать объект как стрелку «Управление». Я использую стрелку «Вход».
3. Регламенты не показываются как стрелки «Управление». Мы не моделируем в рамках «производственной» функции, как сотрудники читают регламенты. Сотрудник к моменту выполнения процесса должен быть уже подготовлен к работе и знать их. Чтение регламентов происходит либо при сборке Операционной или Обслуживающей системы в Системе развития), либо при приеме новых сотрудников на место уволенных в Обслуживающей системе.
Если очень нужно, регламент вместе с другими справочными документами можно приложить к описанию процесса в среде моделирования.

Класс стрелок «Механизмы»

Использование стрелок «Механизмы» только когда это реально необходимо для отображения состава подсистемы, реализующей функцию. В BS все средства деятельности, включая персонал, можно указать в свойствах функции и разгрузить схему.

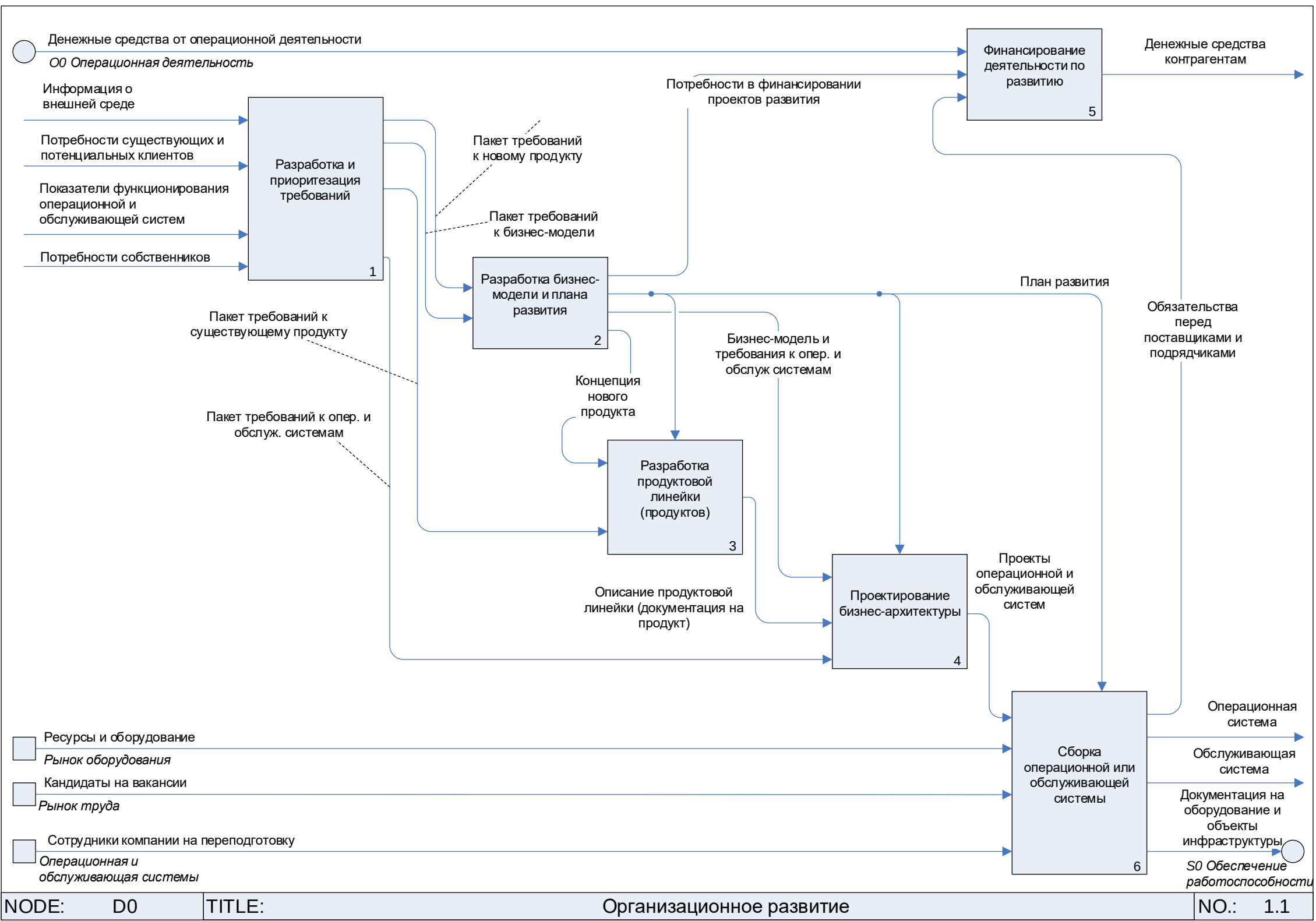
Необходимость связи Выход-Механизм возникает, когда подсистема собирается ситуативно «по потребности» для реализации конкретной задачи, а потом разбирается. Например, для проекта.

В регулярной же деятельности, в большинстве случаев, подсистема для реализации функции уже собрана:

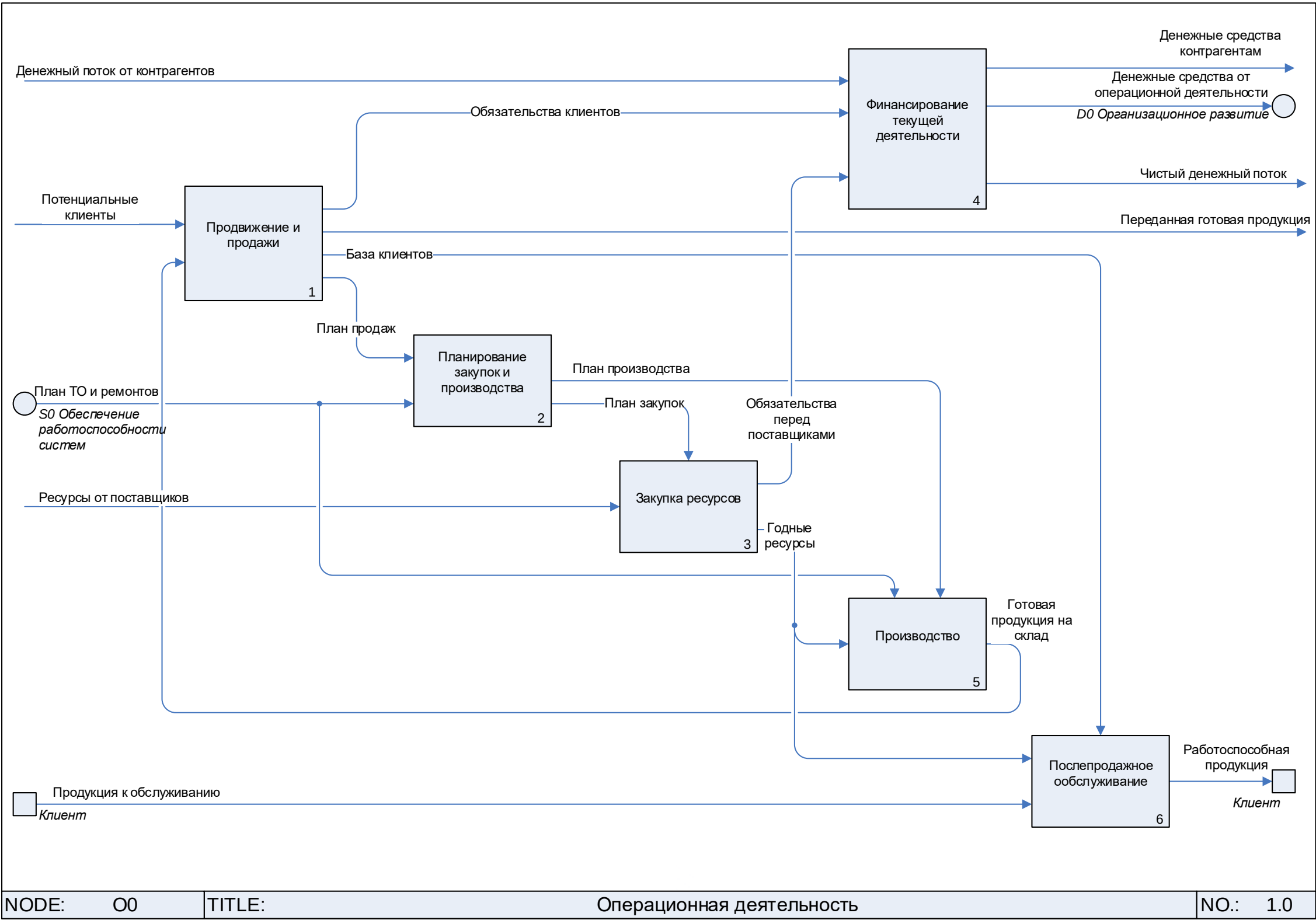
- оборудование закуплено и смонтировано;
- персонал нанят и обучен;
- информационные системы внедрены.

В предлагаемой концепции это делается в Системе развития, а ремонтируются или заменяются части системы в Обслуживающей системе.

Система развития

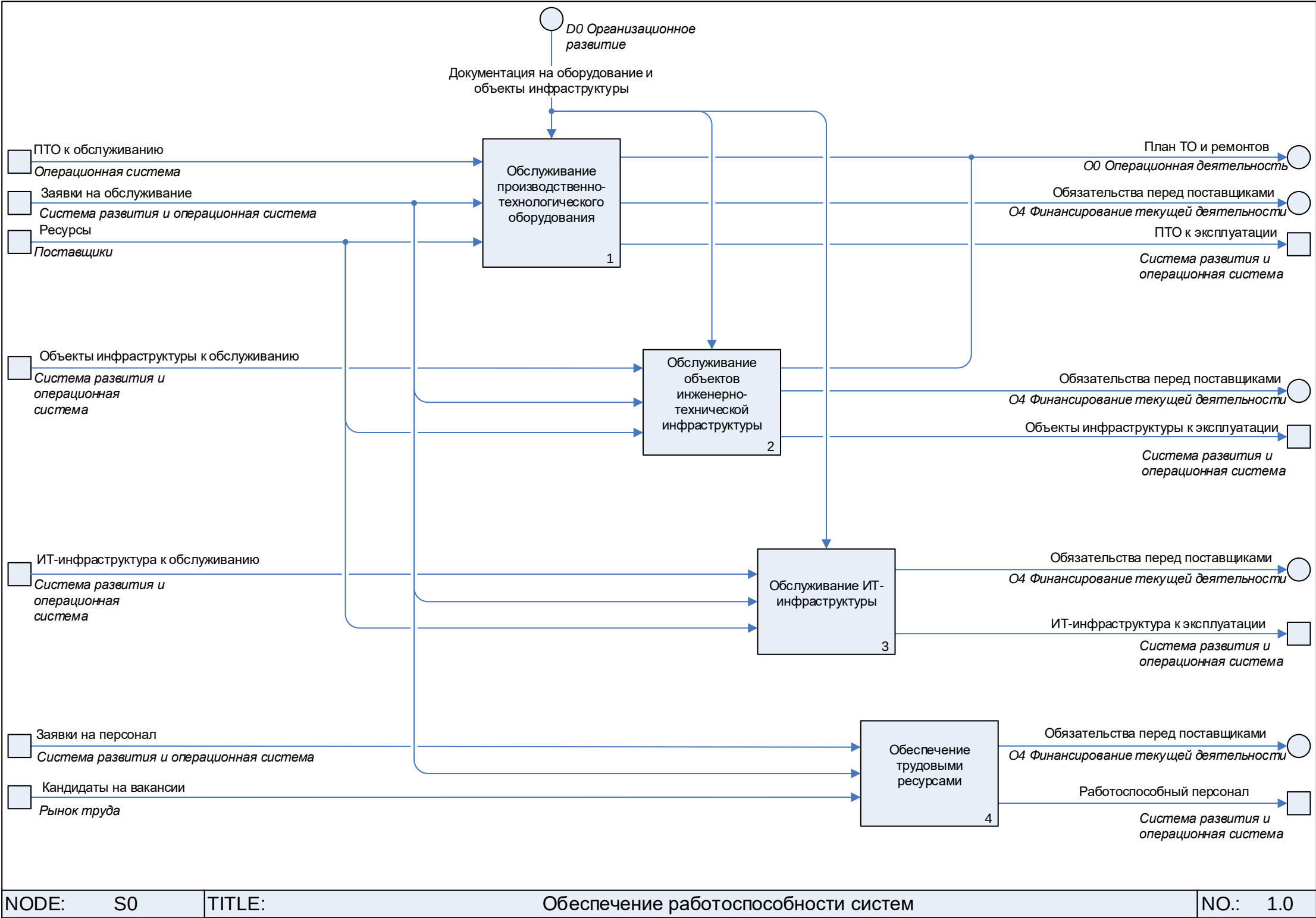


Операционная система



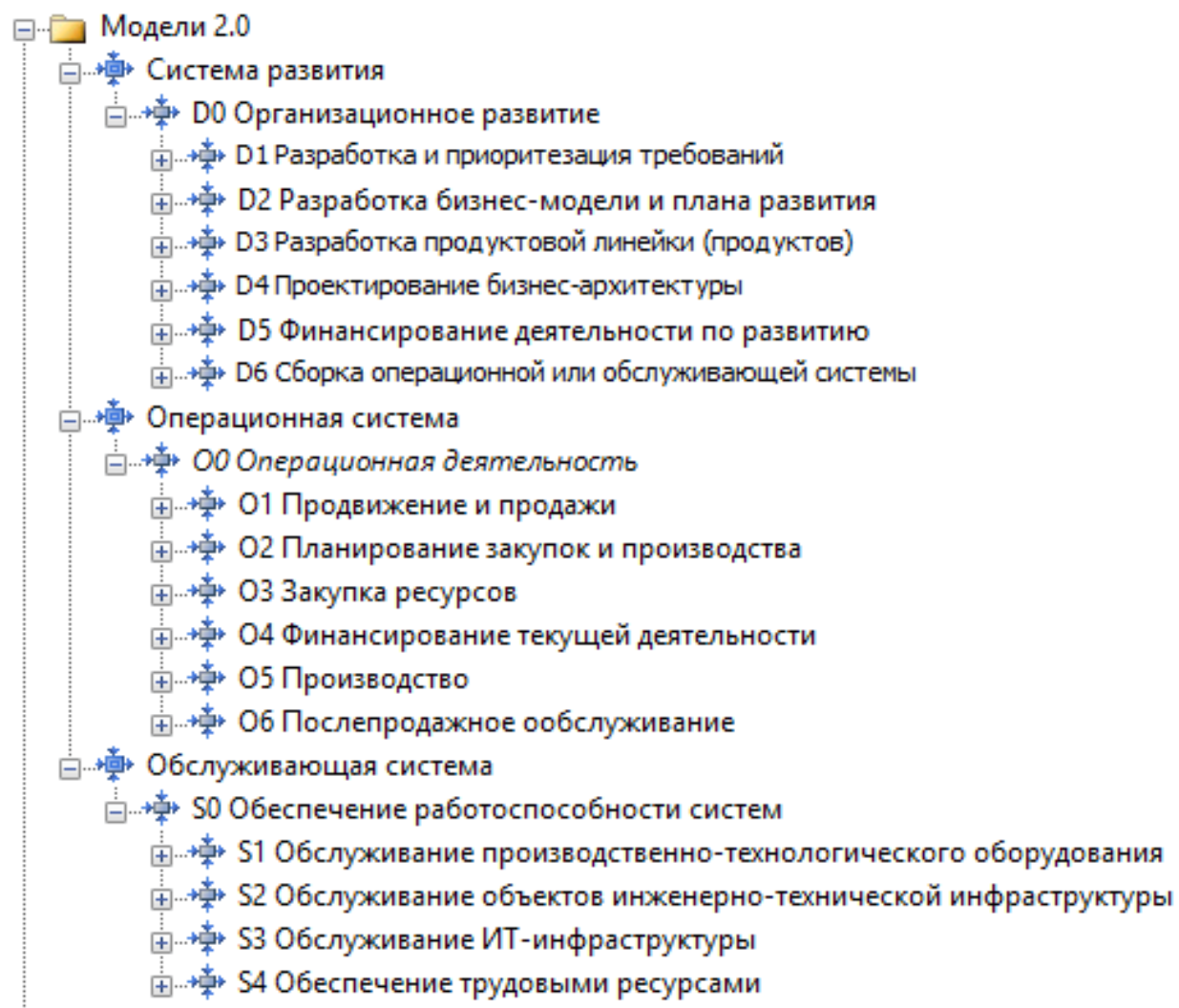


Обслуживающая система





Системный фреймворк модели верхнего уровня



Особенности техники моделирования отдельных систем:

1. Использование междиаграммных ссылок между моделями систем
2. Использование единых «сервисных» функций
3. Избегаем разрыва логической цепочки функций с помощью типовых моделей



Спасибо за внимание!

Вопросы?

Хотите использовать фреймворк?
Напишите мне: dpinaev@businessstudio.ru