



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**"МИРЭА - Российский технологический университет"**

**РТУ МИРЭА**

---

Институт информационных технологий (ИТ)  
Кафедра практической и прикладной информатики (ППИ)

**Доклад**

**На тему «Понятие алгоритма и его роль в моделировании процессов»  
по дисциплине «Анализ и концептуальное моделирование систем»**

Выполнили студентки  
группы ИНБО-10-23

29 «мая» 2025 г.

Мешкова А.К.  
Гаврилина А.П.

Москва 2025

**Алгоритм** – это конечная последовательность операций, описывающая, как преобразовать входные данные в выходные и приводящая к решению задачи. Для бизнес-аналитика это означает, что качество диаграмм UML прямо пропорционально качеству описанного алгоритма: чем полнее и однозначнее алгоритм, тем легче обнаружить слабые места, заложить метрики и вывести процесс на уровень автоматизации.

### **Базовые свойства алгоритма**

Алгоритм обладает следующими свойствами:

1. Дискретность (Алгоритм состоит из отдельных шагов, которые выполняются последовательно.)
2. Определенность/детерминированность (Каждый шаг алгоритма должен быть однозначно определен, то есть не допускаются команды, которые можно трактовать по-разному.)
3. Понятность (Язык, на котором написан алгоритм должен быть понятен исполнителю.)
4. Конечность/результативность (Алгоритм завершается за конечное число шагов.)
5. Массовость (Алгоритм должен быть применим к целому классу однотипных задач, а не только к конкретному набору входных данных.)
6. Эффективность (Алгоритм должен решать задачу за разумное время и с использованием допустимого объема ресурсов.)

### **Алгоритмы как фундамент процессного мышления**

В бизнес-среде алгоритм превращает расплывчатый «процесс» в повторяющуюся, измеримую цепочку действий. Это критично по трём причинам:

- Унификация: один алгоритм — единый стандарт для всех исполнителей, что снижает вариативность качества.
- Метрики: явные шаги легко «прикрутить» к KPI и SLA, анализируя время, стоимость и узкие места.
- Автоматизация: системы BPM и ERP требуют точного алгоритма.

Отсутствие формализованного алгоритма ведёт к тому, что сотрудники трактуют шаги по-разному, а метрики становятся недостоверны.

## **От алгоритма к визуальной модели UML**

### **Диаграмма активности**

Это прямое графическое отображение алгоритма. Для бизнес-аналитика плюс в том, что один рисунок сразу читают как менеджеры, так и разработчики.

### **Диаграмма последовательности**

Фокус ряда шагов смещён на сообщения между «акторами» и системами. Она полезна, когда алгоритм распределён между сервисами.

### **Диаграмма коммуникаций**

Диаграмма коммуникаций фокусируется на том, кто с кем взаимодействует, а не на временной последовательности. Она показывает роли объектов и их связи, помогая проверить, все ли интерфейсы учтены и нет ли лишних зависимостей. Это упрощает анализ структуры взаимодействий.

### **Диаграмма классов**

Алгоритм опирается на данные; диаграмма классов фиксирует «статический скелет» этих данных — классы, атрибуты, операции и связи. Привязав каждый шаг диаграммы активности к методу класса, легко убедиться, что у алгоритма есть доступ ко всем нужным сущностям.

### **Диаграмма анализа классов**

На ранней стадии классы ещё не привязаны к конкретному коду; диаграмма анализа формирует семантическую карту предметной области, отделяя понятия бизнеса от технической реализации. Связывая шаги алгоритма с «диаграммой анализа классов», аналитик демонстрирует заказчику, что логика опирается на правильно выделенные бизнес-сущности.

### **Диаграмма компонентов**

Когда алгоритм готов к развёртыванию, компонентная диаграмма показывает, как блоки кода (компоненты) и их интерфейсы реализуют ранее описанные шаги.

Эти шесть диаграмм вместе покрывают поведение, структуру и реализацию: начиная с процедурной логики (диаграмма активности), проходя сквозь взаимодействия (диаграммы последовательности и коммуникаций), закрепляя их на данных (диаграмма класса) и заканчивая физической раскладкой (диаграммой компонентов).

### **Проверка корректности алгоритма на диаграмме**

1. Все базовые шаги выполнимы без зацикливания.
2. При обработке ошибок нужно следить, чтобы в каждом условном ответвлении код либо полностью выполнялся и завершался, либо сразу возвращал управление вызывающему. Если после проверки остаются «висящие» операции без явного завершения или возврата, это может привести к непредвиденным ошибкам.
3. Каждое требование должно ссылаться на конкретный узел диаграммы; так легче менять алгоритм без «побочных эффектов».

### **Итоги**

Алгоритм объединяет теорию вычислений, менеджмент качества и автоматизацию: без него нельзя ни доказать корректность решения, ни автоматизировать процесс. UML-диаграммы полезны ровно в той мере, в какой точен лежащий в их основе алгоритм. Лучше начинать с четкого текстового описания алгоритма, а уже затем переходить к UML. Также необходимо проверять диаграммы на соответствие ключевым свойствам. В таком случае модель останется управляемой даже после множества итераций доработок.