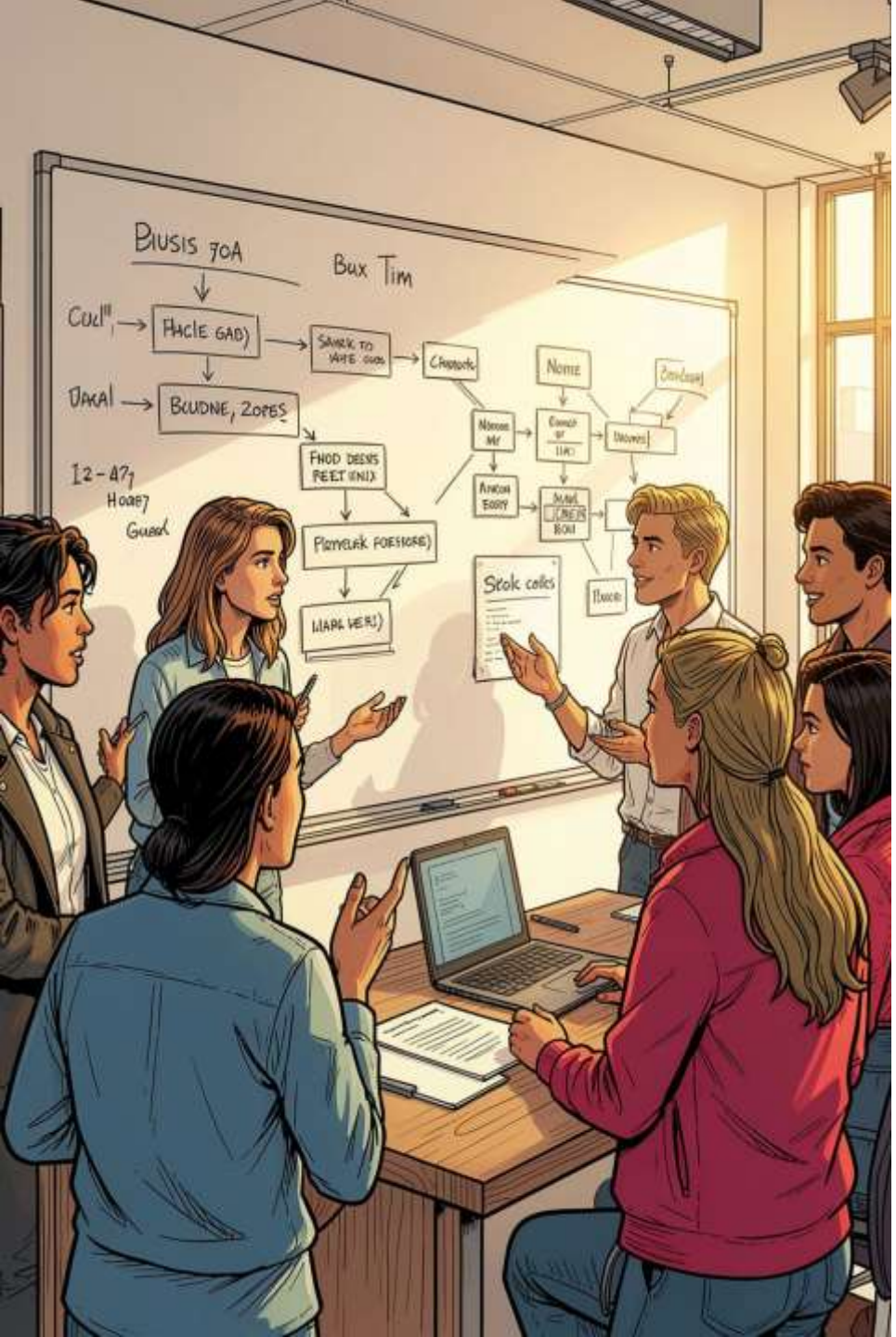
АО «АЛМАТИНСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»
https://atu.edu.kz/

Факультет «Информационные технологии»
Кафедра «Компьютерная инженерия»

Название курса: «Инструментальные средства разработки программ»



Зиятбекова Гулзат Зиятбеккызы – PhD, ассоциированный профессор
g.ziyatbekova@atu.edu.kz



Функциональное моделирование: DFD и IDEF0

Краткий обзор и практическое руководство по моделированию функций систем двумя популярными методиками: диаграммами потоков данных (DFD) и функциональным моделированием IDEF0. Цель презентации — объяснить когда и как применять каждую методику, показать ключевые элементы и привести наглядные примеры.



Зачем нам нужно функциональное моделирование?

Функциональное моделирование помогает понять, как система работает «с точки зрения функций» — какие процессы выполняются, какие данные используются и куда они направляются. Модели упрощают коммуникацию между аналитиками, разработчиками и заказчиками, снижают риски ошибок в требованиях и служат основой для проектирования, тестирования и оптимизации.

Единый язык

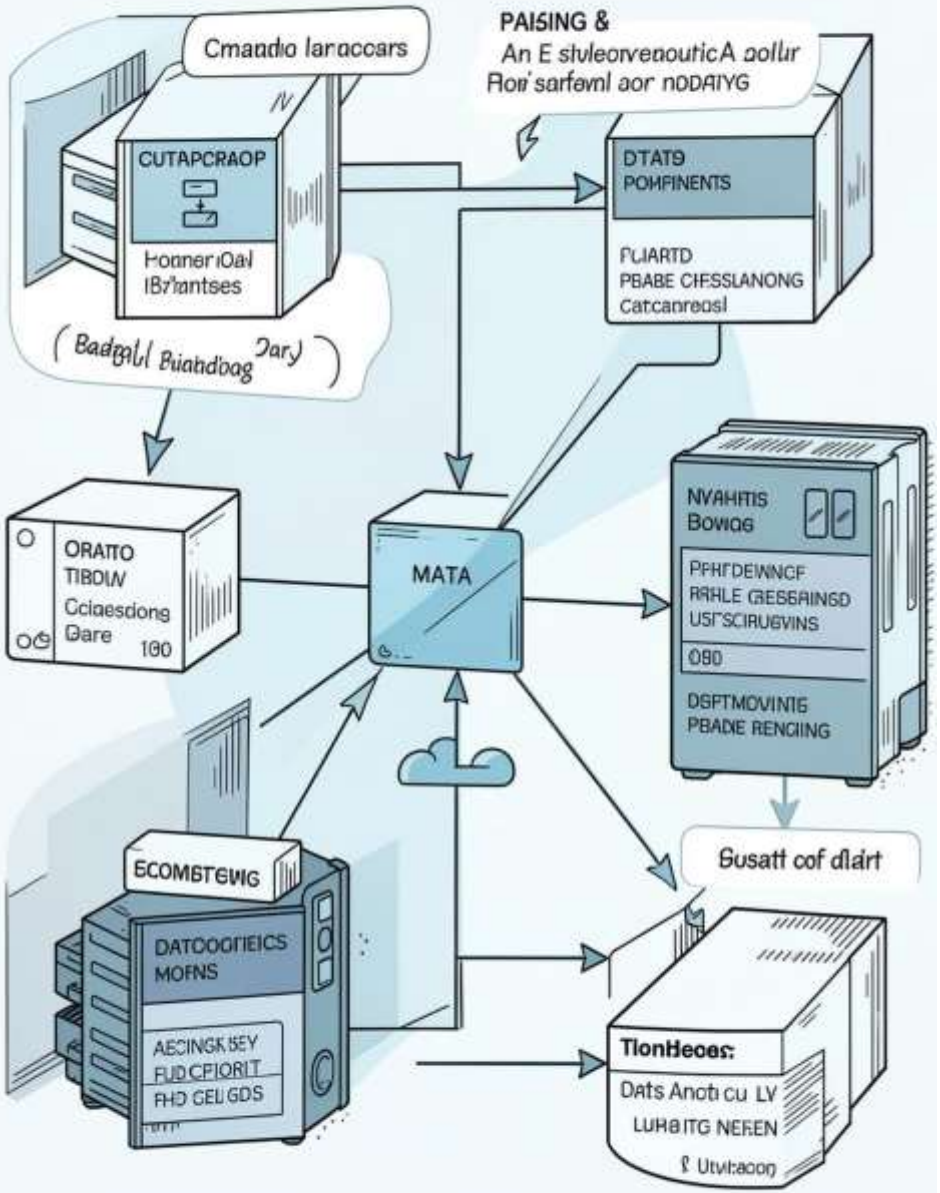
Обеспечивает понятную визуализацию для всех участников проекта.

Снижение рисков

Раннее выявление противоречий и упущенных требований.

Оптимизация

Анализ узких мест и возможностей улучшения процессов.



Введение в методику DFD (Data Flow Diagram)

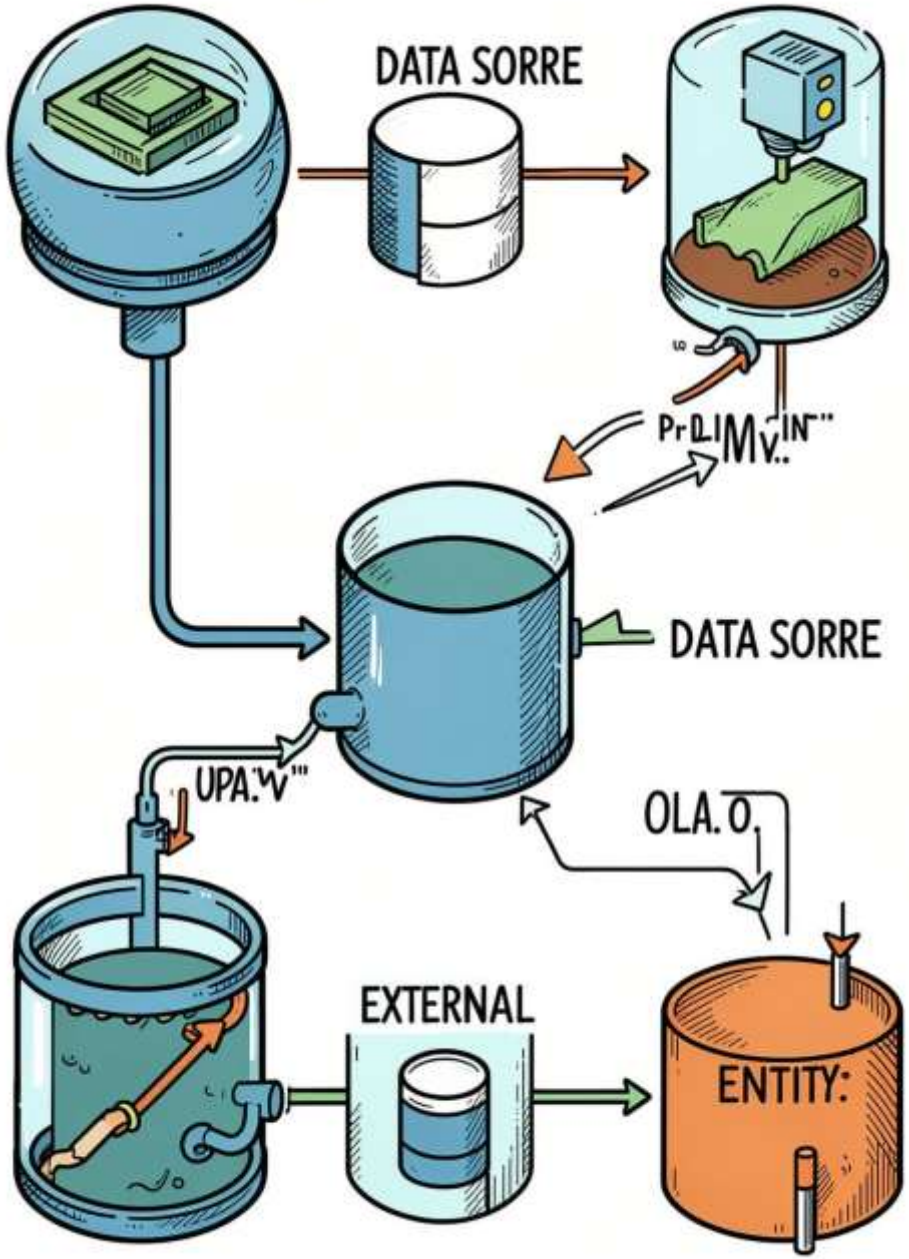
DFD — это графическая нотация для описания потоков данных внутри информационной системы. Главное — показать, откуда приходят данные, какие процессы их преобразуют и где они хранятся. DFD фокусируется на информационных связях, не вдаваясь в реализацию (т.е. без подробностей о программном коде или инфраструктуре).

Ключевая идея

Моделирование потоков данных для понимания функциональной структуры системы.

Уровни детализации

Контекстная диаграмма → уровень 0 → уровни 1..n (декомпозиция процессов).



Основные элементы DFD

Процессы

Действия, которые преобразуют входящие данные в выходные. Обозначаются кругами или прямоугольниками с закруглёнными краями; имеют уникальные номера и имена.

Хранилища данных

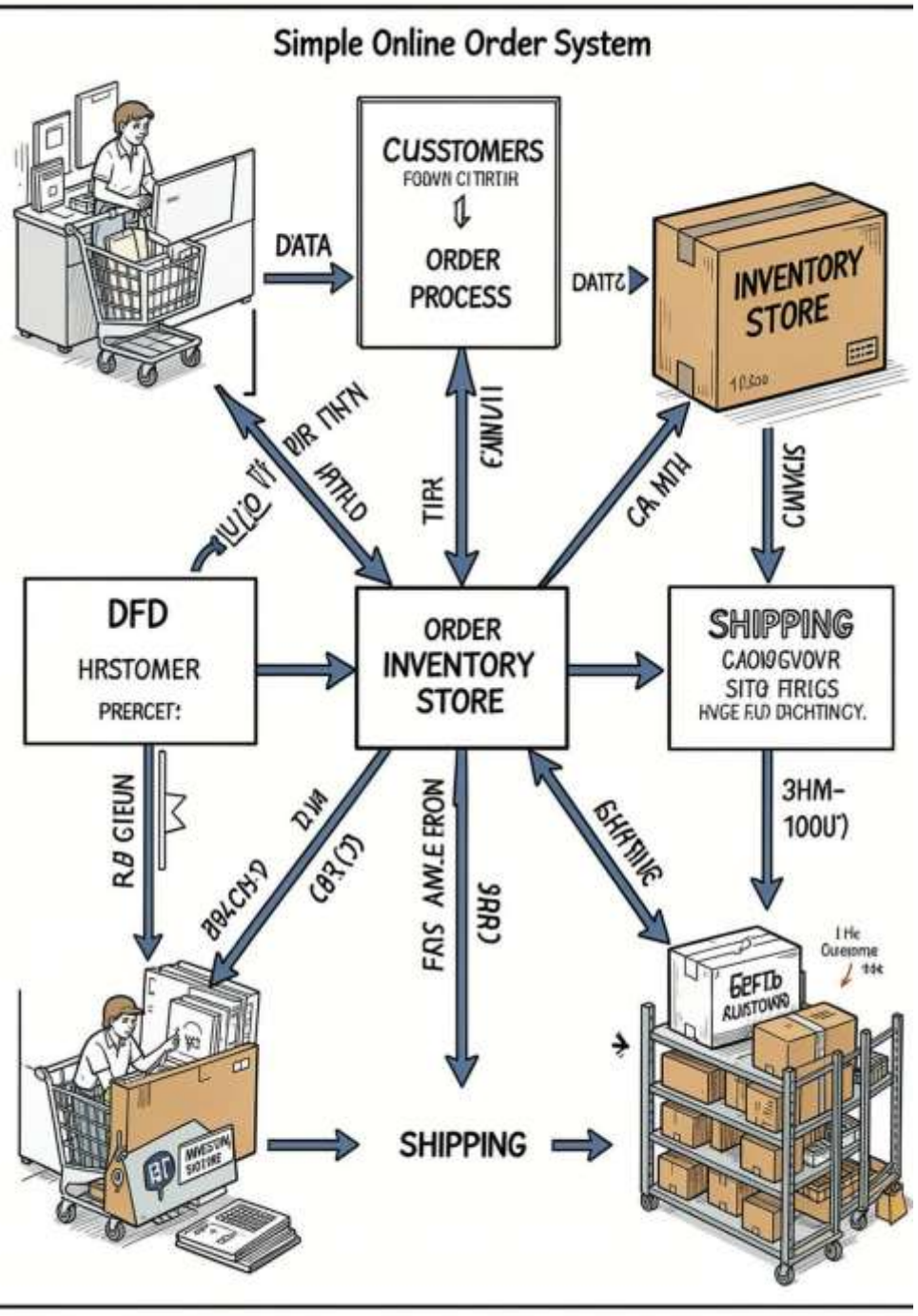
Места постоянного или временного хранения информации (файлы, БД). Обозначаются как параллельные линии или прямоугольники.

Внешние сущности

Внешние к системе участники: пользователи, другие системы, организации — источники и потребители данных.

Потоки данных

Направленные соединения, показывающие движение данных между элементами; должны быть подписаны (какие данные передаются).



Пример построения DFD для простой системы

Рассмотрим интернет-магазин: внешняя сущность — Покупатель; процессы — Принятие заказа, Обработка платежа, Комплектация; хранилище — Каталог/Склад; потоки данных — Заказ, Платёж, Статус доставки. Начинаем с контекстной диаграммы (система как единый процесс), затем декомпозируем на уровень 0 и 1, уточняя названия процессов и форматы потоков данных.

Шаг 1

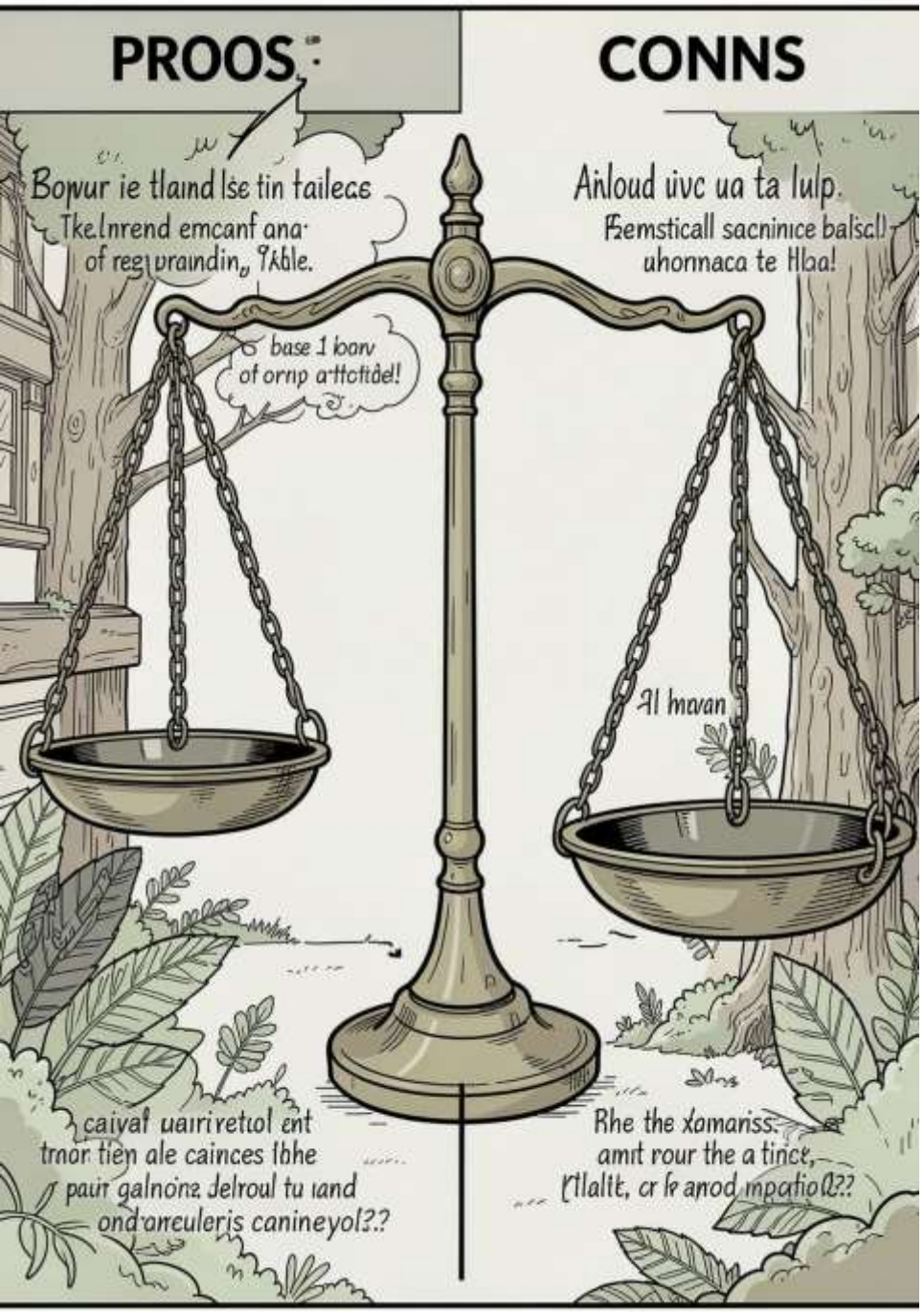
Контекстная диаграмма: показать внешние сущности и основные потоки.

Шаг 2

Уровень 0: выделить ключевые процессы внутри системы.

Шаг 3

Декомпозиция: развернуть сложные процессы на уровни 1..n.



Ограничения и преимущества DFD

Преимущества

- Ясное представление потоков данных
- Простота на ранних этапах анализа
- Подходит для коммуникации с заказчиком

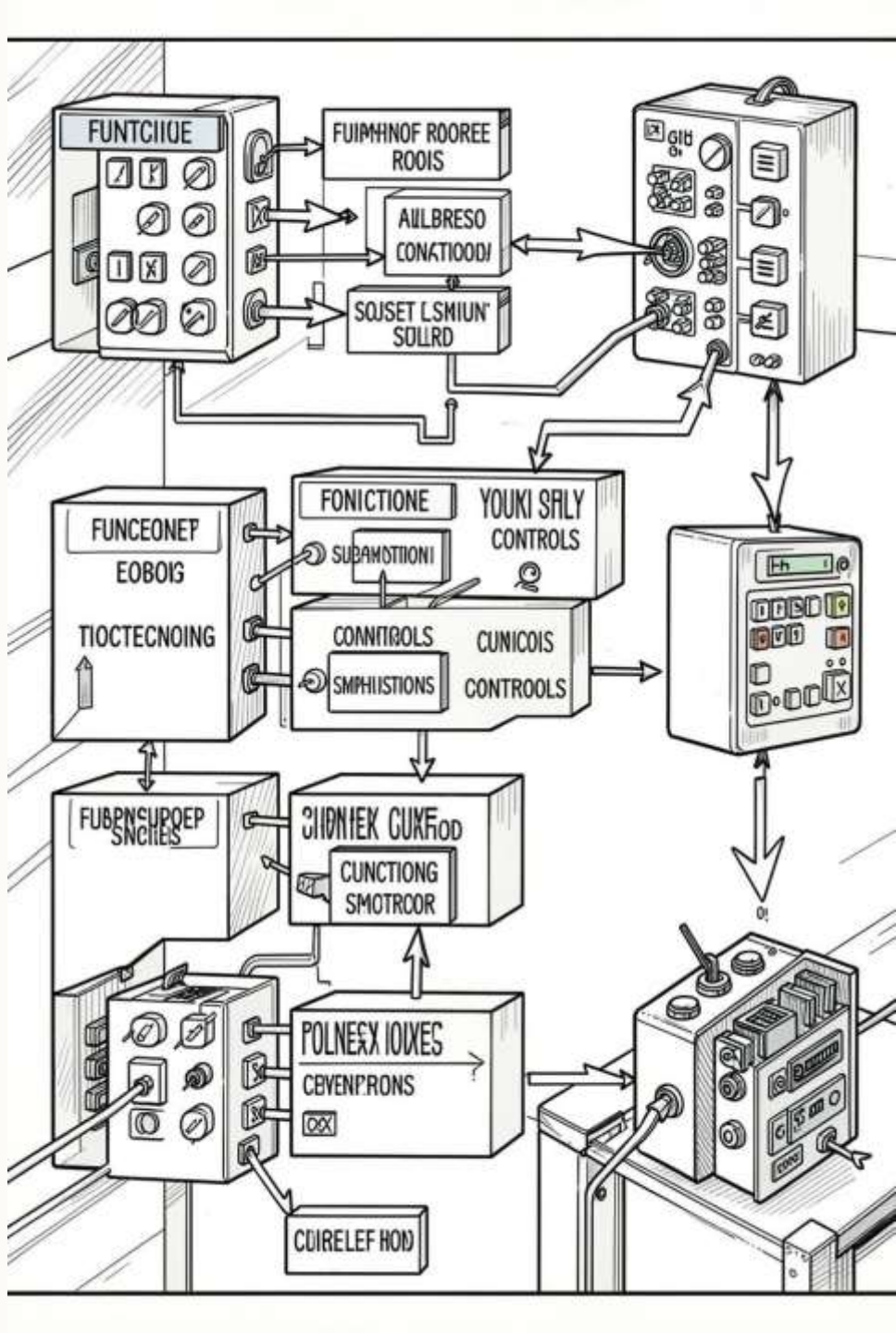
Ограничения

- Не описывает управление и ресурсы
- Может быть неоднозначной при плохой нотации
- Требует дисциплины при декомпозиции (консистентность имен)

Рекомендация: использовать DFD для анализа информационных потоков и интегрировать с другими моделями (процедурными диаграммами, спецификациями интерфейсов) при необходимости.

Введение в методику IDEF0

IDEF0 — методология функционального моделирования, ориентированная на представление функций (действий) и связей управления, ресурсов и входов/выходов. Основная цель — формализовать «что делает система», «чем она управляется» и «какие механизмы используются». IDEF0 удобен для описания сложных организационных и технических систем на разных уровнях детализации.



Основные элементы IDEF0



Блоки (A0, A1...)

Представляют функции или действия — обычно прямоугольники с идентификатором и кратким названием функции.



Входы

Данные или материалы, необходимые функции для производства выходов (располагаются слева).



Выходы

Результаты выполнения функции (располагаются справа).



Управления (Control)

Политики, правила, инструкции, ограничения — поступают сверху и задают условия выполнения.



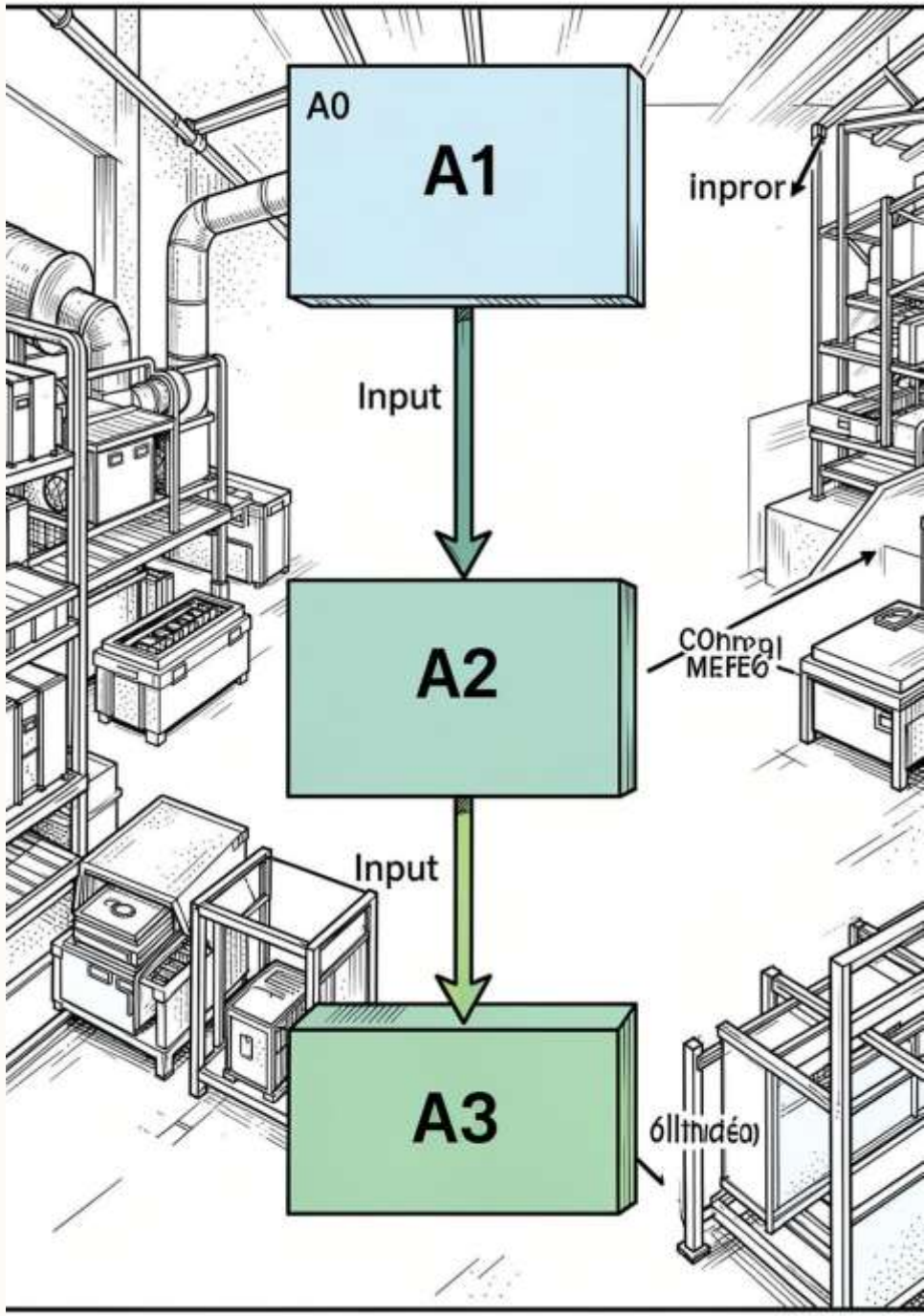
Механизмы

Ресурсы и средства (люди, оборудование, ПО), обеспечивающие выполнение функции (располагаются снизу).

Пример построения IDEF0 диаграммы

Возьмём процесс обработки заказа: верхний блок A0 — «Обработка заказа». Входы: Заказ клиента; Выходы: Подтверждение, Накладные; Управления: Политика ценообразования, SLA; Механизмы: Система ERP, оператор. Декомпозиция: A1 — Проверка заказа, A2 — Расчёт цены, A3 — Передача на склад. На следующих уровнях уточняются подфункции и необходимые ресурсы.

Совет: поддерживайте согласованность управляющих сигналов при декомпозиции — это облегчает проверку правильности модели.



Сравнение DFD и IDEF0: когда что использовать

DFD — когда использовать

Лучше для моделирования информационных потоков и взаимодействия между подсистемами. Удобен при проектировании интерфейсов, интеграции и анализа данных.

IDEF0 — когда использовать

Подходит для описания функций, их управления и ресурсов на организационном или системном уровне, особенно когда важно показать механизмы и правила.

Ключевые выводы: эти методы дополняют друг друга — DFD про данные и процессы, IDEF0 про функции, управление и механизмы. Рекомендуется использовать их совместно: DFD для детализации информационных потоков внутри функций, IDEF0 — для структурирования функций и управляющих ограничений.

Рекомендуемая практика: начать с IDEF0 для целостного представления функций и затем создавать DFD для критичных процессов, где важны форматы и маршруты данных.

