

СИСТЕМНАЯ ПОДГОТОВКА СПЕЦИАЛИСТОВ ПО DATA SCIENCE: ОБРАЗОВАТЕЛЬНАЯ АРХИТЕКТУРА ПЕРЕХОДА ОТ LOW-CODE К FULL-CODE

Кондрашова Е.С., Миловидова А.А.

Федеральное государственное бюджетное образовательное учреждение высшего образования «Университет «Дубна», kes.th@uni-dubna.ru

Спрос на специалистов Data Science растет на 25–30% ежегодно. Традиционная модель на базе Python создает высокие барьеры входа и когнитивную перегрузку, низкоуровневые платформы обеспечивают быстрые результаты, но ограничены в гибкости. Возникает противоречие: как сочетать оба подхода? Большинство программ выбирают либо теорию, либо практику, не интегрируя их в единую архитектуру.

Методика опирается на теорию когнитивной нагрузки, показывающую, что low-code среды снижают нагрузку на синтаксис, и принцип зоны ближайшего развития, подтверждающий, что инструменты расширяют возможности студентов. Спиральная модель предполагает возвращение к материалу с возрастающей сложностью.

Предлагается архитектура с тремя уровнями компетенций: визуальный, гибридный и программный. Ключевым принципом является инвариантность логики анализа — одна задача имеет эквивалентное представление в низкоуровневом и полнокодовом форматах. Это укрепляет понимание и сокращает время формирования базовых компетенций на 30–40%.

Архитектура включает пять компонентов. Целевой определяет компетенции: аналитическое мышление, подготовка данных, исследовательский анализ, моделирование и интерпретация. Содержательный организует материал по спиральной модели. Технологический задает последовательность: Orange, затем KNIME или LogiDom, затем Python. Организационный структурирует занятия по циклу анализа: постановка задачи, исследование, подготовка данных, моделирование, оценка и интерпретация. Оценочный реализует три уровня: формативная оценка, контрольные точки и портфолио проектов.

Переход происходит через пять фаз: Фаза 1 — 100% low-code, Фаза 2 — 75% low-code и 25% код, Фаза 3 — паритет подходов, Фаза 4 — преобладание кода, Фаза 5 — полный переход к full-code разработке на Python.

Практическая реализация строится на двойственном представлении: каждая задача сначала решается в low-code среде, затем воспроизводится на Python, демонстрируя инвариантность логики и помогая перейти к программному мышлению.

Архитектура разрешает главное противоречие Data Science образования: сочетание быстрого получения практических результатов с глубокими программными навыками. Подход применим к университетским программам и корпоративному обучению.