Профессиональная магистерская специализация

Программная инженерия и качество программного обеспечения

tech технологический университет

tech технологический университет

Профессиональная магистерская специализация

Программная инженерия и качество программного обеспечения

- Формат: **онлайн**
- Продолжительность: **2 года**
- Учебное заведение: **TECH Технологический университет**
- Расписание: **по своему усмотрению**
- Экзамены: **онлайн**

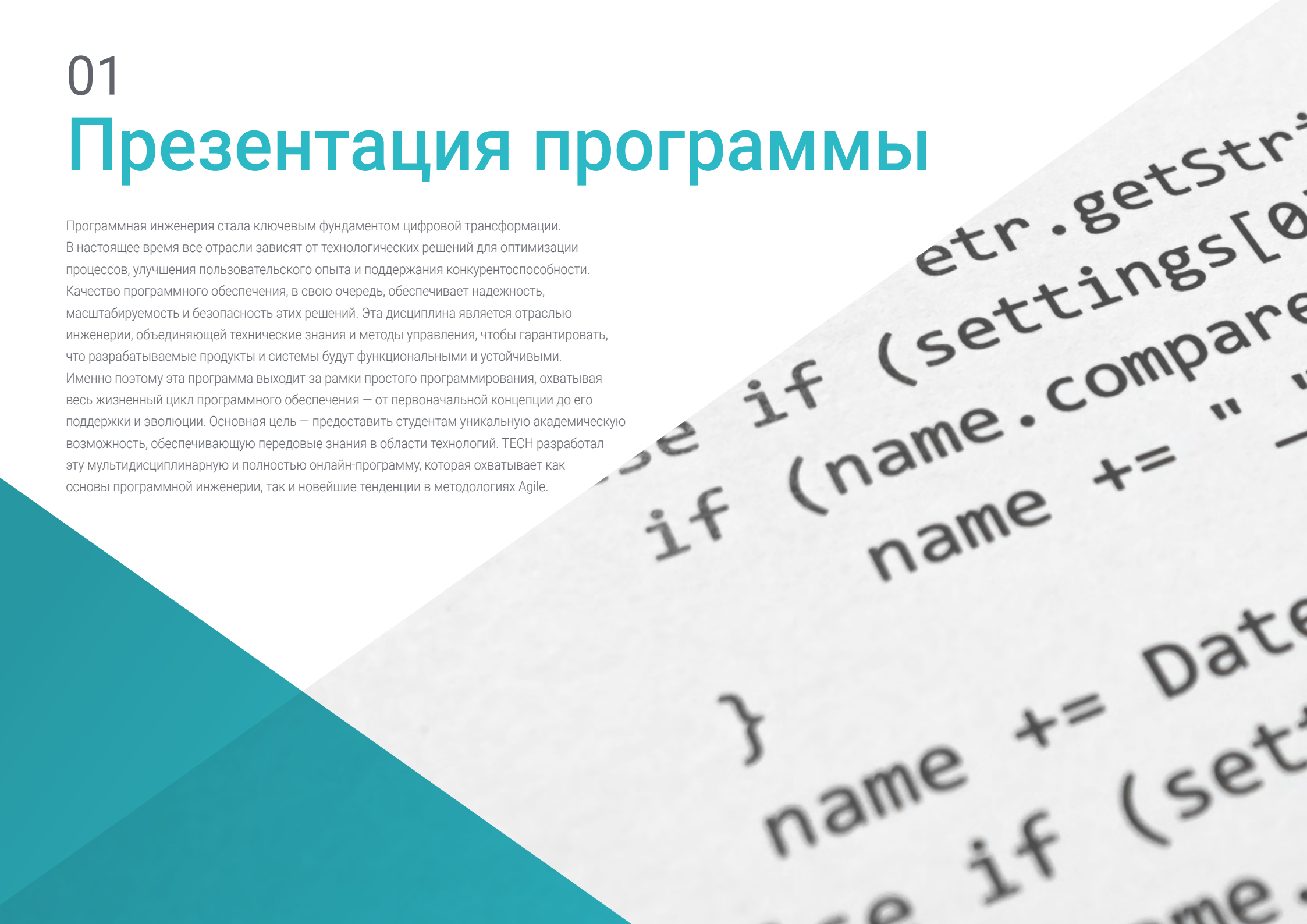
Веб-доступ: www.techtitute.com/ru/information-technology/advanced-master-degree/advanced-master-degree-software-engineering-quality

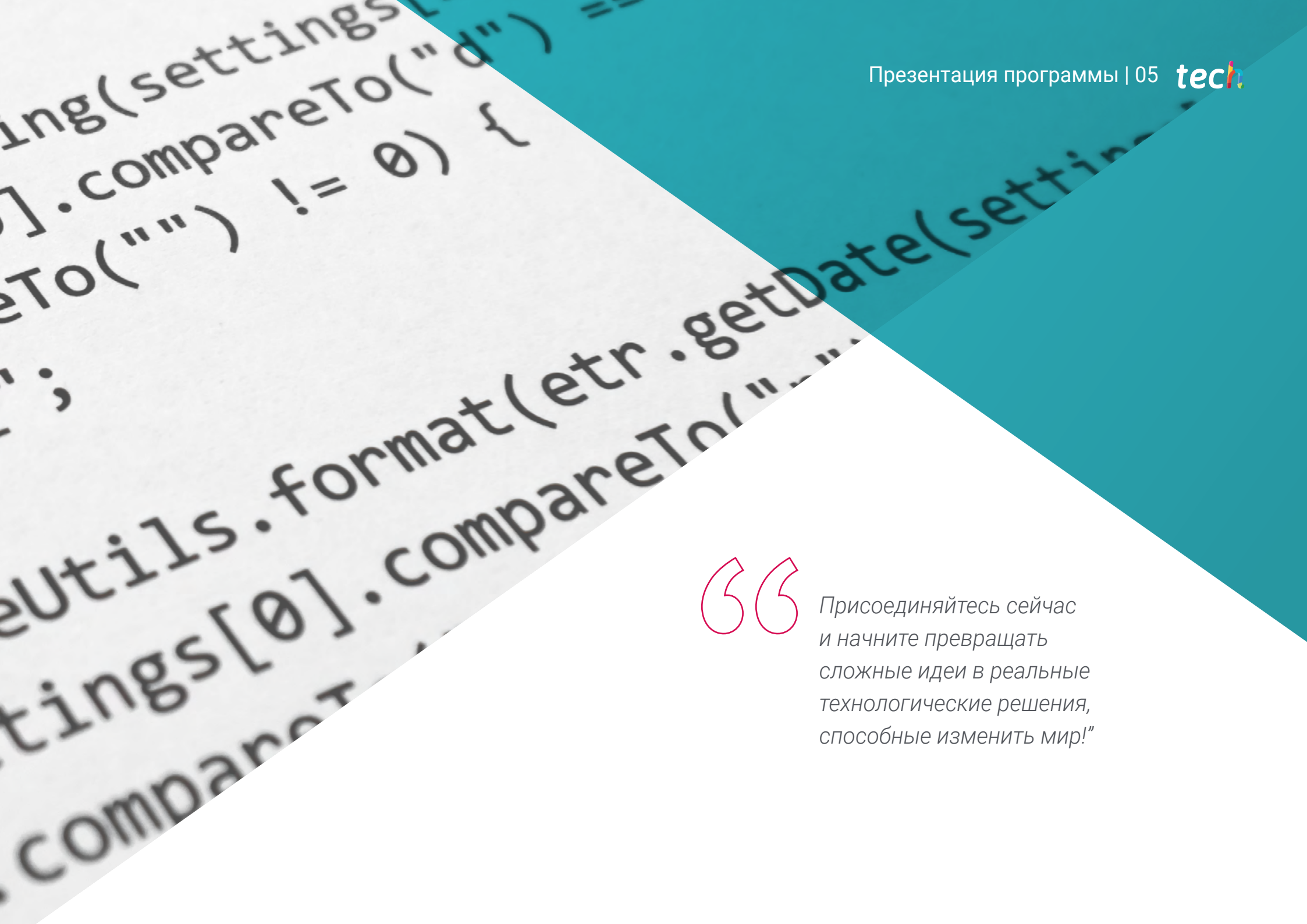
Оглавление

01

Презентация программы

Программная инженерия стала ключевым фундаментом цифровой трансформации. В настоящее время все отрасли зависят от технологических решений для оптимизации процессов, улучшения пользовательского опыта и поддержания конкурентоспособности. Качество программного обеспечения, в свою очередь, обеспечивает надежность, масштабируемость и безопасность этих решений. Эта дисциплина является отраслью инженерии, объединяющей технические знания и методы управления, чтобы гарантировать, что разрабатываемые продукты и системы будут функциональными и устойчивыми. Именно поэтому эта программа выходит за рамки простого программирования, охватывая весь жизненный цикл программного обеспечения — от первоначальной концепции до его поддержки и эволюции. Основная цель — предоставить студентам уникальную академическую возможность, обеспечивающую передовые знания в области технологий. TECH разработал эту мультидисциплинарную и полностью онлайн-программу, которая охватывает как основы программной инженерии, так и новейшие тенденции в методологиях Agile.





Присоединяйтесь сейчас и начните превращать сложные идеи в реальные технологические решения, способные изменить мир!”

Качество программного обеспечения гарантирует, что системы не только соответствуют функциональным требованиям, но и являются интуитивно понятными, безопасными и устойчивыми в долгосрочной перспективе. Этот аспект особенно важен в критически значимых секторах, таких как финансы, здравоохранение и транспорт, где сбои могут привести к серьезным последствиям. Кроме того, ориентация на качество позволяет компаниям гибко адаптироваться к постоянному технологическому прогрессу и эффективно реагировать на растущие требования рынка.

С помощью таких методологий, как гибкая разработка, *DevOps* и внедрение международных стандартов качества, инженерия программного обеспечения гарантирует поставку продуктов в более короткие сроки. Кроме того, контролируемые затраты и уровень качества, минимизирующий критические ошибки, были улучшены за счет интеграции новых технологий, таких как искусственный интеллект, *облачные вычисления* и кибербезопасность. В этом контексте программа, разработанная TECH, направлена на подготовку высококвалифицированных специалистов по проектированию, разработке, управлению и обеспечению качества программного обеспечения. Для приобретения необходимых навыков учебный план Профессиональной магистерской специализации включает самые актуальные концепции по управлению технологическими проектами и стратегическому руководству. Этот подход представляет собой дополнительную ценность как для инженеров, которые уже занимают ответственные должности и хотят обновить свои знания, так и для тех, кто впервые стремится возглавить команды и проекты в этой области.

Одним из основных преимуществ этой программы является то, что она будет проводиться полностью онлайн, что исключает необходимость в поездках и привязке к конкретным расписаниям. Кроме того, студенты смогут воспользоваться методом обучения *Relearning*, который адаптируется к их индивидуальному темпу учебы. Этот гибкий подход оказывается очень полезным, так как позволяет студентам эффективно организовывать свои повседневные обязательства — будь то профессиональные или семейные — и таким образом достичь полного развития.

Данная **Профессиональная магистерская специализация в области программной инженерии и качества программного обеспечения** содержит самую полную и актуальную программу на рынке. Основными особенностями обучения являются:

- Разбор практических кейсов, представленных экспертами в области информатики
- Наглядное, схематичное и исключительно практическое содержание курса предоставляет научную и практическую информацию по тем дисциплинам, которые необходимы для осуществления профессиональной деятельности
- Практические упражнения для самопроверки, контроля и улучшения успеваемости
- Особое внимание уделяется инновационным методикам в области программной инженерии и качества программного обеспечения
- Теоретические занятия, вопросы эксперту, дискуссионные форумы по спорным темам и самостоятельная работа
- Учебные материалы курса доступны с любого стационарного или мобильного устройства с выходом в интернет

В TECH вы научитесь не только разрабатывать программное обеспечение, но и создавать системы, которые меняют жизнь людей и компаний к лучшему"

Овладейте самыми передовыми инженерными навыками и инструментами с помощью самой инновационной методики преподавания на современной академической сцене"

Повысьте свои профессиональные ожидания, обучаясь на 100% онлайн, не мешая своим личным и семейным обязанностям.

Станьте ведущим инженером-профессионалом, готовым учиться из любой точки мира.

В преподавательский состав программы входят профессионалы в области информатики, которые вносят свой опыт работы в эту программу, а также признанные специалисты, принадлежащие к ведущим научным сообществам.

Мультимедийное содержание программы, разработанное с использованием новейших образовательных технологий, позволит специалисту пройти обучение с учетом ситуации и контекста, то есть в интерактивной среде, которая обеспечит погружение в учебный процесс, запрограммированный на обучение в реальных ситуациях.

В центре внимания этой программы — проблемно-ориентированное обучение, с помощью которого студент должен попытаться разрешить различные ситуации из профессиональной практики, возникающие в течение учебного курса. Для этого специалисту будет помогать инновационная интерактивная видеосистема, созданная признанными и опытными специалистами.

02

Почему стоит учиться в TECH?

TECH – крупнейший в мире цифровой университет. Имея впечатляющий каталог из более чем 14 000 академических программ, доступных на 11 языках, он позиционируется как лидер по трудоустройству с показателем 99%. Кроме того, университет располагает огромным преподавательским составом, включающим более 6 000 преподавателей с высочайшим международным авторитетом.

Пройдите обучение в крупнейшем в мире цифровом университете и обеспечьте себе профессиональный успех. Будущее начинается в TECH”

Лучший онлайн-университет в мире по версии FORBES

Авторитетный журнал Forbes, специализирующийся на бизнесе и финансах, отметил TECH как "лучший онлайн-университет в мире" Об этом недавно сообщили в статье цифровой версии издания, где рассматривается успешный кейс этого учебного заведения, "благодаря его академическому предложению, отбору преподавательского состава и инновационному методу обучения, ориентированному на подготовку профессионалов будущего"

Лучший международный преподавательский состав

Преподавательский состав TECH включает более 6 000 специалистов с мировым признанием. Среди профессоров, исследователей и топ-менеджеров транснациональных корпораций — Исайя Ковингтон, тренер "Бостон Селтикс", Магда Романска, главный исследователь Harvard MetaLAB, Игнасио Вистумба, председатель отделения трансляционной молекулярной патологии в MD Anderson Cancer Center, Д.У. Пайн, креативный директор журнала TIME и другие.

Крупнейший цифровой университет в мире

TECH – крупнейший в мире цифровой университет. Мы — крупнейшее образовательное учреждение с самым обширным цифровым каталогом учебных программ, полностью онлайн, охватывающим большинство областей знаний. Мы предлагаем самое большое количество программ с выдачей дипломов собственного образца, а также официальных программ бакалавриата и программ последипломной подготовки в мире. В общей сложности более 14 000 университетских программ на десяти языках, что делает нас крупнейшим образовательным учреждением в мире.

Самые полные учебные программы в университетской среде

TECH предлагает наиболее полные учебные программы, охватывающие как фундаментальные концепции, так и ключевые научные достижения в каждой конкретной области. Кроме того, эти программы постоянно обновляются, чтобы обеспечить студентам передовое академическое образование и наиболее востребованные профессиональные навыки. Таким образом, программы TECH дают студентам значительное преимущество для успешного карьерного роста.

Уникальный метод обучения

TECH — первый университет, использующий метод *Relearning* во всех своих учебных программах. Это лучшая методология онлайн-обучения, сертифицированная международными агентствами образовательного качества. Кроме того, эта инновационная академическая модель дополняется "Методом кейсов", формируя уникальную стратегию онлайн-обучения. В программу также включены передовые учебные ресурсы, среди которых подробные видеоматериалы, инфографики и интерактивные конспекты.

Официальный онлайн-университет NBA

TECH — официальный онлайн-университет NBA. Благодаря нашему партнерству с крупнейшей баскетбольной лигой мы предлагаем студентам эксклюзивные образовательные программы, а также широкий спектр учебных материалов, посвященных бизнесу лиги и другим аспектам спортивной индустрии. Каждая программа имеет уникальный учебный план и включает выдающихся приглашенных лекторов — профессионалов с выдающейся спортивной карьерой, которые поделятся своим опытом по самым актуальным темам.

Лидеры по трудоустройству

TECH удалось стать университетом-лидером по трудоустройству. 99% студентов получают работу по специальности в течение одного года после окончания любой из программ университета. Столько же студентов сразу же добиваются карьерного роста. Все это благодаря методологии обучения, эффективность которой основана на приобретении практических навыков, необходимых для профессионального развития.

Google Partner Premier

Американский технологический гигант присвоил TECH знак Google Partner Premier. Эта награда, доступная лишь 3% компаний мира, подчеркивает эффективный, гибкий и адаптированный подход, который этот университет предоставляет своим студентам. Признание не только подтверждает высокий уровень строгости, производительности и инвестиций в цифровую инфраструктуру TECH, но и ставит этот университет среди ведущих технологических компаний мира.

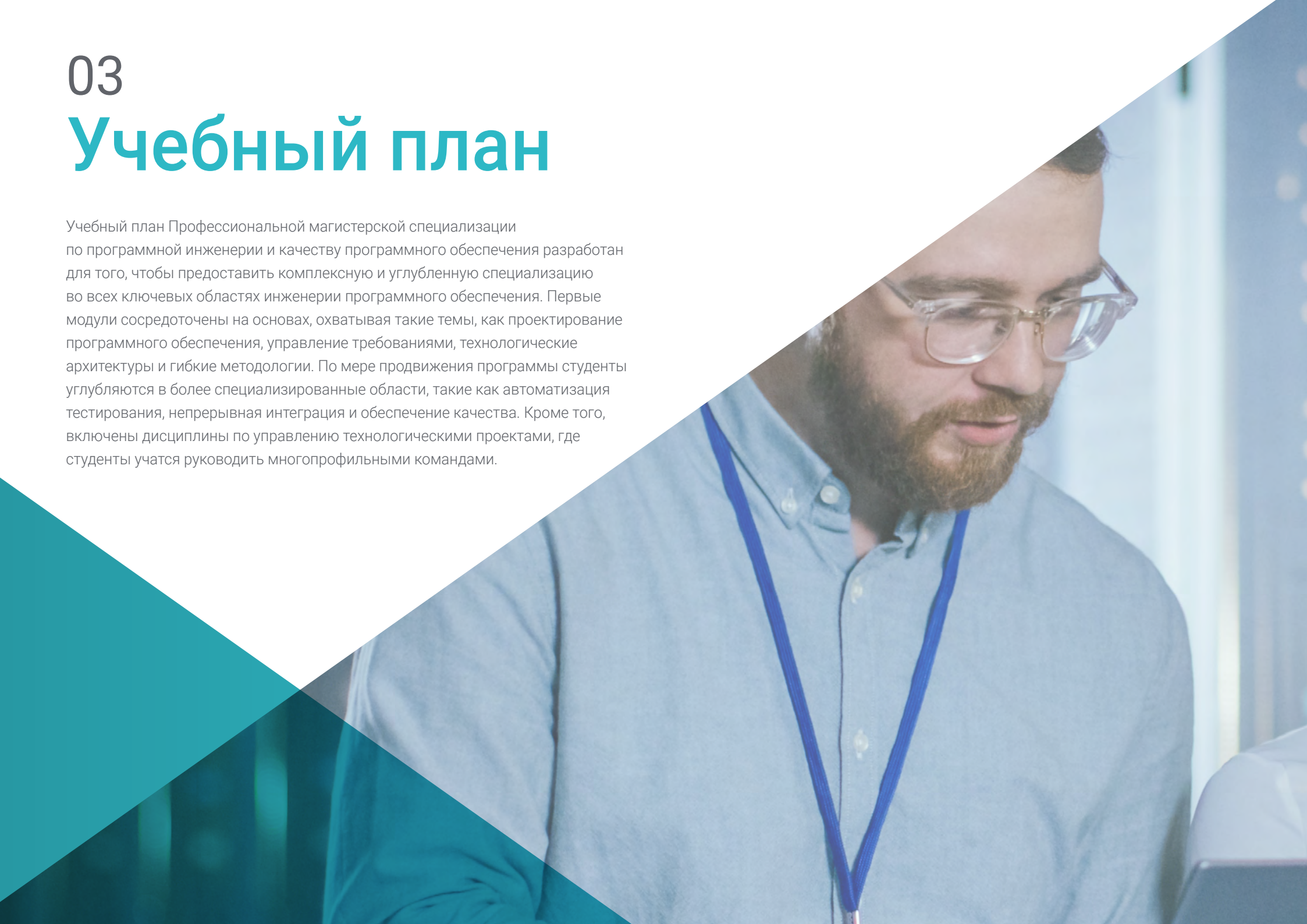
Университет, получивший самые высокие оценки от своих студентов

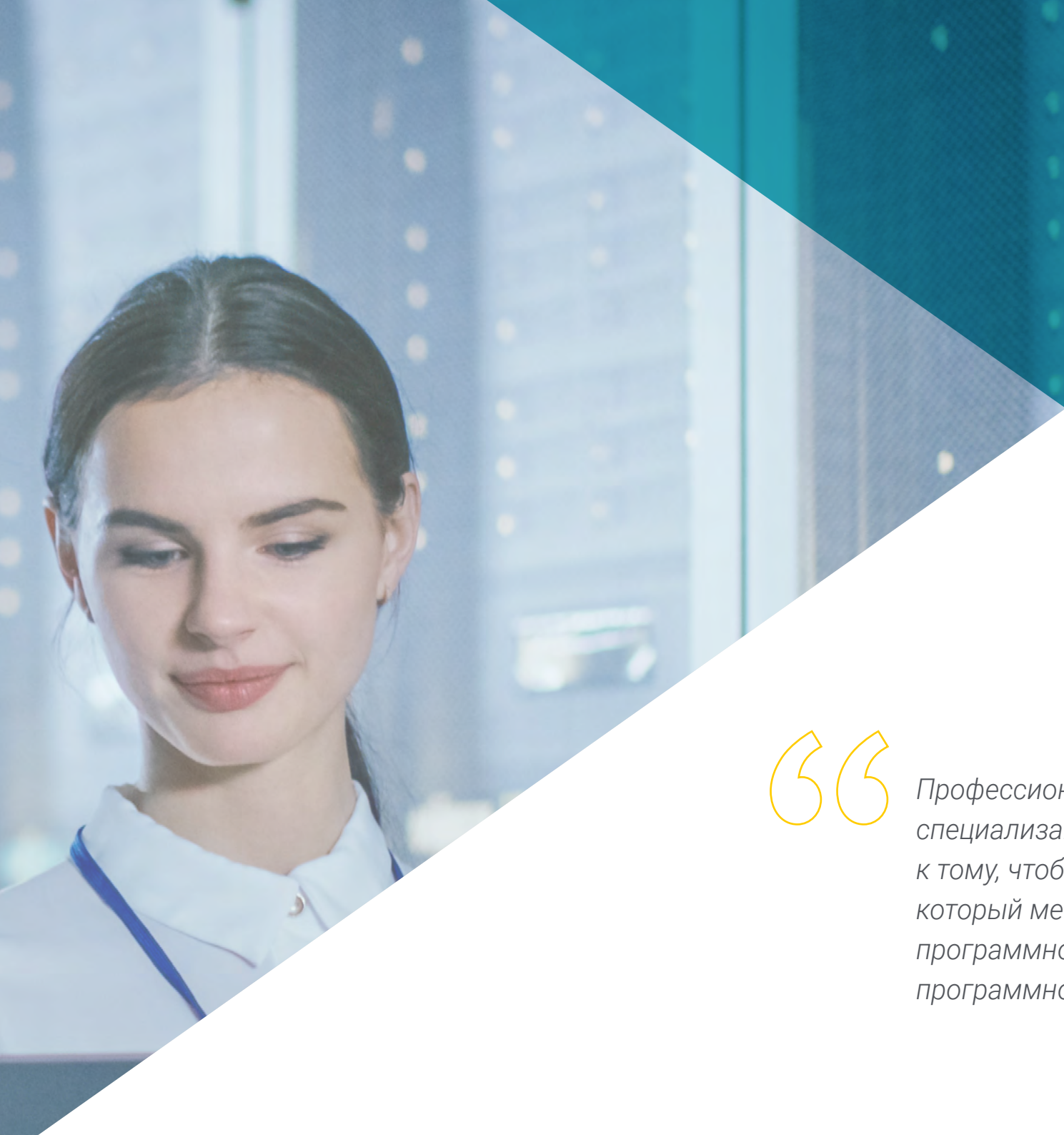
Студенты признали TECH самым высоко оцененным университетом в мире на ведущих платформах с отзывами, отметив его высший рейтинг — 4,9 из 5, основанный на более чем 1 000 рецензиях. Эти результаты укрепляют позиции TECH как ведущего международного университета, отражая его превосходство и положительное влияние образовательной модели. Оценки, которые закрепляют за TECH статус абсолютного эталона среди университетов на международном уровне.

03

Учебный план

Учебный план Профессиональной магистерской специализации по программной инженерии и качеству программного обеспечения разработан для того, чтобы предоставить комплексную и углубленную специализацию во всех ключевых областях инженерии программного обеспечения. Первые модули сосредоточены на основах, охватывая такие темы, как проектирование программного обеспечения, управление требованиями, технологические архитектуры и гибкие методологии. По мере продвижения программы студенты углубляются в более специализированные области, такие как автоматизация тестирования, непрерывная интеграция и обеспечение качества. Кроме того, включены дисциплины по управлению технологическими проектами, где студенты учатся руководить многопрофильными командами.





Профессиональная магистерская специализация готовит вас к тому, чтобы стать экспертом, который меняет ситуацию в сфере программной инженерии и качества программного обеспечения"

Модуль 1. Качество программного обеспечения. Уровни развития TRL

1.1. Элементы, влияющие на качество программного обеспечения (I). Технический долг
- 1.1.1. Технический долг. Причины и последствия
- 1.1.2. Качество программного обеспечения. Общие принципы
- 1.1.3. Беспринципное и принципиальное качество программного обеспечения
  - 1.1.3.1. Последствия
  - 1.1.3.2. Необходимость применения принципов качества в программном обеспечении
- 1.1.4. Качество программного обеспечения. Типология
- 1.1.5. Качество программного обеспечения. Специфические особенности

1.2. Элементы, влияющие на качество программного обеспечения (II). Сопутствующие расходы
- 1.2.1. Качество программного обеспечения. Влияющие элементы
- 1.2.2. Качество программного обеспечения. Заблуждения
- 1.2.3. Качество программного обеспечения. Сопутствующие расходы

1.3. Модели качества программного обеспечения (I). Управление знаниями
- 1.3.1. Общие модели качества
  - 1.3.1.1. Всеобщее управление качеством
  - 1.3.1.2. Европейская модель делового совершенства (EFQM)
  - 1.3.1.3. Концепция шести сигм
- 1.3.2. Модели управления знаниями
  - 1.3.2.1. Модель Dyba
  - 1.3.2.2. Модель Seks
- 1.3.3. Фабрика опыта и парадигма QIP
- 1.3.4. Качество используемых моделей (25010)

1.4. Модели качества программного обеспечения (III). Качество данных, процессов и моделей SEI
- 1.4.1. Модель качества данных
- 1.4.2. Моделирование процессов программного обеспечения
- 1.4.3. Software & *Systems Process Engineering Metamodel Specification* (SPEM)
- 1.4.4. Модели SEI
  - 1.4.4.1. CMMI
  - 1.4.4.2. SCAMPI
  - 1.4.4.3. IDEAL

1.5. Стандарты качества программного обеспечения ISO (I). Анализ стандартов
- 1.5.1. Стандарт ISO 9000
  - 1.5.1.1. Стандарт ISO 9000
  - 1.5.1.2. Семейство стандартов качества ISO (9000)
- 1.5.2. Другие стандарты ISO, связанные с качеством
- 1.5.3. Стандарты моделирования качества (ISO 2501)
- 1.5.4. Стандарты измерения качества (ISO 2502n)

1.6. Стандарты качества программного обеспечения ISO (II). Требования и оценка
- 1.6.1. Стандарты по требованиям к качеству (2503n)
- 1.6.2. Стандарты по оценке качества (2504n)
- 1.6.3. ISO/IEC 24744:2007

1.7. Уровни развития TRL (I). Уровни с 1 по 4
- 1.7.1. Уровни TRL
- 1.7.2. Уровень 1: основные принципы
- 1.7.3. Уровень 2: концепция и/или применение
- 1.7.4. Уровень 3: критическая аналитическая функция
- 1.7.5. Уровень 4: проверка основных компонентов в лабораторных условиях

1.8. Уровни развития TRL (II). Уровни с 5 по 9
- 1.8.1. Уровень 5: проверка основных компонентов в реальных условиях.
- 1.8.2. Уровень 6: модель системы/подсистемы
- 1.8.3. Уровень 7: демонстрация в условиях эксплуатации
- 1.8.4. Уровень 8: полная и сертифицированная система
- 1.8.5. Уровень 9: успех в реальных условиях

1.9. Уровни развития TRL. Использование
- 1.9.1. Пример компании с лабораторной средой
- 1.9.2. Пример компании, занимающейся НИОКР
- 1.9.3. Пример промышленной компании, занимающейся НИОКР
- 1.9.4. Пример совместной лабораторно-инженерной компании

1.10. Качество программного обеспечения. Ключевые детали
- 1.10.1. Методологические детали
- 1.10.2. Технические детали
- 1.10.3. Детали управления программными проектами
  - 1.10.3.1. Качество информационных систем
  - 1.10.3.2. Качество программного продукта
  - 1.10.3.3. Качество программного обеспечения

Модуль 2. Разработка программных проектов. Функциональная и техническая документация.

2.1. Управление проектами
- 2.1.1. Управление проектами в области качества программного обеспечения
- 2.1.2. Управление проектами. Преимущества
- 2.1.3. Управление проектами. Типология

2.2. Методика управления проектами
- 2.2.1. Методика управления проектами
- 2.2.2. Проектные методики. Типология
- 2.2.3. Методики управления проектами. Применение

2.3. Фаза выявления требований
- 2.3.1. Определение требований к проекту
- 2.3.2. Управление проектными совещаниями
- 2.3.3. Предоставляемая документация

2.4. Модель
- 2.4.1. Начальный этап
- 2.4.2. Этап анализа
- 2.4.3. Этап строительства
- 2.4.4. Этап тестирования
- 2.4.5. Предоставление

2.5. Используемая модель данных
- 2.5.1. Определение новой модели данных
- 2.5.2. Определение плана миграции данных
- 2.5.3. Набор данных

2.6. Влияние на другие проекты
- 2.6.1. Влияние проекта. Примеры

2.7. *MUST* проекта
- 2.7.1. *MUST* проекта
- 2.7.2. Идентификация *MUST* проекта
- 2.7.3. Определение точек реализации проекта

2.8. Команда для строительства проекта
- 2.8.1. Роли, которые необходимо выполнять в соответствии с проектом
- 2.8.2. Контакт с отделом кадров для подбора персонала
- 2.8.3. Результаты и график проекта

2.9. Технические аспекты программного проекта
- 2.9.1. Архитектор проекта. Технические аспекты
- 2.9.2. Технические лидеры
- 2.9.3. Проектирование программного обеспечения
- 2.9.4. Оценка качества кода Sonar

2.10. Результаты проекта
- 2.10.1. Функциональный анализ
- 2.10.2. Модель данных
- 2.10.3. Диаграмма состояний
- 2.10.4. Техническая документация

Модуль 3. *Тестирование* программного обеспечения. Автоматизация тестирования

3.1. Модели качества программного обеспечения
- 3.1.1. Качество продукта
- 3.1.2. Качество процесса
- 3.1.3. Качество использования

3.2. Качество процесса
- 3.2.1. Качество процесса
- 2.3.2. Модели зрелости
- 3.2.3. Стандарт ISO 15504
  - 3.2.3.1. Цели
  - 3.2.3.2. Контекст
  - 3.2.3.3. Этапы

3.3. Стандарт ISO/IEC 15504
- 3.3.1. Категории процессов
- 3.3.2. Процесс разработки. Пример
- 3.3.3. Фрагмент программы
- 3.3.4. Этапы

3.4. CMMI (*Интеграция модели зрелости возможностей*)
- 3.4.1. CMMI. Интеграция модели зрелости возможностей
- 3.4.2. Модели и области. Типология
- 3.4.3. Области процесса
- 3.4.4. Уровни способности
- 3.4.5. Управление процессами
- 3.4.6. Управление проектами

3.5. Управление изменениями и репозиториями
- 3.5.1. Управление изменениями в программном обеспечении
  - 3.5.1.1. Элемент конфигурации. Непрерывная интеграция
  - 3.5.1.2. Линии
  - 3.5.1.3. Блок-схемы
  - 3.5.1.4. *Бранчи*
- 3.5.2. Репозиторий
  - 3.5.2.1. Контроль версий
  - 3.5.2.2. Работа в команде и использование репозитория
  - 3.5.2.3. Непрерывная интеграция в репозитории

3.6. *Team Foundation Server* (TFS)
- 3.6.1. Установка и настройка
- 3.6.2. Создание командного проекта
- 3.6.3. Добавление содержимого в систему управления источниками
- 3.6.4. *TFS в облаке*

3.7. *Тестирование*
- 3.7.1. Мотивы для тестирования
- 3.7.2. Проверочные испытания
- 3.7.3. Бета-тестирование
- 3.7.4. Реализация и обслуживание

3.8. Нагрузочное тестирование
- 3.8.1. *Нагрузочное тестирование*
- 3.8.2. Тестирование *LoadView*
- 3.8.3. Тестирование с помощью K6 *Cloud*
- 3.8.4. Тестирование с помощью *Loader*

3.9. Нагрузочные и выносливые испытания устройства
- 3.9.1. Мотивы для проведения модульного тестирования
- 3.9.2. Инструменты для *модульного тестирования*
- 3.9.3. Мотивы для проведения стресс-тестирования
- 3.9.4. *Стресс-тестирование*
- 3.9.5. Мотивы для проведения тестов на устойчивость
- 3.9.6. Тестирование с помощью *LoadRunner*

3.10. Масштабируемость. Масштабируемое проектирование программного обеспечения
- 3.10.1. Масштабируемость и архитектура программного обеспечения
- 3.10.2. Независимость между слоями
- 3.10.3. Связь между слоями. Архитектурные паттерны

Модуль 4. Методологии управления проектами программного обеспечения. Каскадная модель vs agile-методологии

4.1. Каскадная модель
- 4.1.1. Каскадная модель
- 4.1.2. Каскадная модель. Влияние на качество программного обеспечения
- 4.1.3. Каскадная модель. Примеры

4.2. Методологии Agile
- 4.2.1. Методологии Agile
- 4.2.2. Agile-методологии. Влияние на качество программного обеспечения
- 4.2.3. Agile-методологии. Примеры

4.3. Методология SCRUM
- 4.3.1. Методология SCRUM
- 4.3.2. SCRUM-манифест
- 4.3.3. Применения SCRUM

4.4. Канбан-доска
- 4.4.1. Канбан-метод
- 4.4.2. Канбан-доска
- 4.4.3. Канбан-доска Примеры применения

4.5. Управление проектами в каскадной модели
- 4.5.1. Фазы проекта
- 4.5.2. Видение в каскадной модели
- 4.5.3. Результаты, которые необходимо рассмотреть

4.6. Управление проектами в SCRUM
- 4.6.1. Фазы проекта SCRUM
- 4.6.2. Видение проекта SCRUM
- 4.6.3. Результаты, которые необходимо рассмотреть

4.7. Каскадная модель vs SCRUM. Сравнение
- 4.7.1. Пилотный проект
- 4.7.2. Проект с применением каскадной модели. Пример
- 4.7.3. Проект с применением SCRUM. Пример

4.8. Видение клиента
- 4.8.1. Документы в каскадной модели
- 4.8.2. Документы в SCRUM
- 4.8.3. Сравнение

4.9. Структура Канбан
- 4.9.1. Истории пользователей
- 4.9.2. Backlog
- 4.9.3. Анализ Канбан

4.10. Гибридные проекты
- 4.10.1. Проектирование
- 4.10.2. Управление проектами
- 4.10.3. Результаты, которые необходимо рассмотреть

Модуль 5. TDD (*Test Driven Developement*). Разработка программного обеспечения через тестирование

5.1. TDD. *Разработка программного обеспечения через тестирование*
- 5.1.1. TDD. *Разработка программного обеспечения через тестирование*
- 5.1.2. TDD. Влияние TDD на качество
- 5.1.3. Проектирование и разработка на основе тестирования. Примеры

5.2. Цикл TDD
- 5.2.1. Выбор требования
- 5.2.2. Тестирование. Типологии
  - 5.2.2.1. Модульные тесты
  - 5.2.2.2. Интеграционное тестирование
  - 5.2.2.3. *Сквозное тестирование*
- 5.2.3. Проверка тестирования. Ошибки
- 5.2.4. Создание реализации
- 5.2.5. Выполнение автоматизированных тестов
- 5.2.6. Устранение дублирования
- 5.2.7. Обновление списка требований
- 5.2.8. Повторение цикла TDD
- 5.2.9. Цикл TDD Теоретический и практический пример

5.3. Стратегии внедрения TDD
- 5.3.1. Ошибочное внедрение
- 5.3.2. Треугольное внедрение
- 5.3.3. Очевидное внедрение

5.4. TDD. Применение. Преимущества и недостатки
- 5.4.1. Преимущества использования
- 5.4.2. Ограничения на использование
- 5.4.3. Баланс качества при реализации

5.5. TDD. Передовая практика
- 5.5.1. Правила TDD
- 5.5.2. Правило 1: Проводить предыдущий неудачный тест, прежде чем приступать к кодированию в производстве.
- 5.5.3. Правило 2: Не писать более одного модульного теста.
- 5.5.4. Правило 3: не писать больше кода, чем требуется
- 5.5.5. Ошибки и антипаттерны, которых следует избегать в TDD

5.6. Моделирование реального проекта для использования TDD (I)
- 5.6.1. Обзор проекта (Компания A)
- 5.6.2. Применение TDD
- 5.6.3. Предлагаемые тесты
- 5.6.4. Тесты. *Обратная связь*

5.7. Моделирование реального проекта для использования TDD (II)
- 5.7.1. Обзор проекта (Компания B)
- 5.7.2. Применение TDD
- 5.7.3. Предлагаемые тесты
- 5.7.4. Тесты. *Обратная связь*

5.8. Моделирование реального проекта для использования TDD (III)
- 5.8.1. Обзор проекта (Компания C)
- 5.8.2. Применение TDD
- 5.8.3. Предлагаемые тесты
- 5.8.4. Тесты. *Обратная связь*

5.9. Альтернативы TDD. *Разработка программного обеспечения через тестирование*
- 5.9.1. TCR (*Test Commit Revert*)
- 5.9.2. BDD (*Behavior Driven Development*)
- 5.9.3. ATDD (*Acceptance Test Driven Development*)
- 5.9.4. TDD. Теоретическое сравнение

5.10. TDD TCR, BDD и ATDD. Практическое сравнение
- 5.10.1. Определение проблемы
- 5.10.2. Решение с помощью TCR
- 5.10.3. Решение с помощью BDD
- 5.10.4. Решение с помощью ATDD

Модуль 6. DevOps Управление качеством программного обеспечения

6.1. DevOps Управление качеством программного обеспечения
- 6.1.1. DevOps
- 6.1.2. DevOps и качество программного обеспечения
- 6.1.3. DevOps. Преимущества культуры DevOps

6.2. DevOps. Отношения с Agile
- 6.2.1. Ускоренная доставка
- 6.2.2. Качество
- 6.2.3. Снижение затрат

6.3. Запуск DevOps
- 6.3.1. Определение проблемы
- 6.3.2. Внедрение в компании
- 6.3.3. Метрики внедрения

6.4. Цикл доставки программного обеспечения
- 6.4.1. Методы проектирования
- 6.4.2. Соглашения
- 6.4.3. План действий

6.5. Разработка кода без ошибок
- 6.5.1. Удобный в обслуживании код
- 6.5.2. Схемы развития
- 6.5.3. *Тестирование* кода
- 6.5.4. Разработка программного обеспечения на уровне кода. Передовая практика

6.6. Автоматизация
- 6.6.1. Автоматизация. Виды тестирования
- 6.6.2. Стоимость автоматизации и технического обслуживания
- 6.6.3. Автоматизация. Смягчение ошибок

6.7. Развертывания
- 6.7.1. Оценка цели
- 6.7.2. Разработка автоматического и адаптированного процесса
- 6.7.3. Обратная связь и оперативность

6.8. Управление инцидентами
- 6.8.1. Готовность к инцидентам
- 6.8.2. Анализ и разрешение инцидентов
- 6.8.3. Как избежать ошибок в будущем

6.9. Автоматизация развертывания
- 6.9.1. Подготовка к автоматическому развертыванию
- 6.9.2. Автоматическая оценка состояния процессов
- 6.9.3. Метрики и возможность отката

6.10. Передовая практика. Эволюция DevOps
- 6.10.1. Руководство по использованию DevOps
- 6.10.2. DevOps Методология для команды
- 6.10.3. Избегание ниш

Модуль 7. DevOps и непрерывная интеграция. Передовые практические решения в разработке программного обеспечения

7.1. Поток поставки программного обеспечения
- 7.1.1. Идентификация действующих лиц и артефактов
- 7.1.2. Проектирование потока поставки программного обеспечения
- 7.1.3. Поток поставки программного обеспечения. Межэтапные требования

7.2. Автоматизация процессов
- 7.2.1. Непрерывная интеграция
- 7.2.2. Непрерывное развертывание
- 7.2.3. Конфигурирование сред и управление секретами

7.3. Декларативные конвейеры
- 7.3.1. Различия между традиционными, кодоподобными и декларативными конвейерами
- 7.3.2. Декларативные конвейеры
- 7.3.3. Декларативный конвейер в Jenkins
- 7.3.4. Сравнение поставщиков услуг непрерывной интеграции

7.4. Качественные ворота и богатая обратная связь
- 7.4.1. Качественные ворота
- 7.4.2. Стандарты качества с качественными воротами. Обслуживание
- 7.4.3. Бизнес-требования в запросах на интеграцию

7.5. Управление артефактами
- 7.5.1. Артефакты и жизненный цикл
- 7.5.2. Системы хранения и управления артефактами
- 7.5.3. Безопасность в управлении артефактами

7.6. Непрерывное развертывание
- 7.6.1. Непрерывное развертывание в виде контейнеров
- 7.6.2. Непрерывное развертывание с помощью PaaS

7.7. Улучшение времени выполнения конвейера: статический анализ и *Git Hooks*
- 7.7.1. Статический анализ
- 7.7.2. Правила стиля кода
- 7.7.3. *Git Hooks* и *модульные тесты*
- 7.7.4. Влияние инфраструктуры

7.8. Уязвимости контейнеров
- 7.8.1. Уязвимости контейнеров
- 7.8.2. Сканирование изображений
- 7.8.3. Периодические отчеты и оповещения

Модуль 8. Проектирование баз данных (БД). Стандартизация и производительность. Качество программного обеспечения

8.1. Проектирование баз данных
- 8.1.1. Базы данных. Типология
- 8.1.2. Используемые в настоящее время базы данных
  - 8.1.2.1. Реляционные
  - 8.1.2.2. Ключ-значение
  - 8.1.2.3. На основе сети
- 8.1.3. Качество данных

8.2. Проектирование модели сущность - связь (I)
- 8.2.1. Модель сущность - связь. Качество и документация
- 8.2.2. Сущности
  - 8.2.2.1. Сильная сущность
  - 8.2.2.2. Слабая сущность
- 8.2.3. Атрибуты
- 8.2.4. Набор отношений
  - 8.2.4.1. 1 к 1
  - 8.2.4.2. 1 к многим
  - 8.2.4.3. Многие к 1
  - 8.2.4.4. Многие ко многим
- 8.2.5. Ключевые моменты
  - 8.2.5.1. Первичный ключ
  - 8.2.5.2. Внешний ключ
  - 8.2.5.3. Первичный ключ слабой сущности
- 8.2.6. Ограничения
- 8.2.7. Кардинальность
- 8.2.8. Наследственность
- 8.2.9. Агрегация

8.3. Модель взаимоотношений между сущностями(II). Инструменты
- 8.3.1. Модель сущность-связь. Инструменты
- 8.3.2. Модель сущность-связь. Наглядный пример
- 8.3.3. Реальная модель сущность-связь
  - 8.3.3.1. Визуальный пример
  - 8.3.3.2. Образец в представлении таблицы

8.4. Стандартизация базы данных (БД) (I). Соображения по качеству программного обеспечения
- 8.4.1. Стандартизация БД и качество
- 8.4.2. Зависимости
  - 8.4.2.1. Функциональная зависимость
  - 8.4.2.2. Свойства функциональной зависимости
  - 8.4.2.3. Предполагаемые свойства
- 8.4.3. Ключевые моменты

8.5. Стандартизация базы данных (БД) (II). Нормальная форма и правила Кодда
- 8.5.1. Нормальные формы
  - 8.5.1.1. Первая нормальная форма (1FN)
  - 8.5.1.2. Вторая нормальная форма (2FN)
  - 8.5.1.3. Третья нормальная форма (3FN)
  - 8.5.1.4. Нормальная форма Бойса — Кодда (BCNF)
  - 8.5.1.5. Четвертая нормальная форма (4FN)
  - 8.5.1.6. Пятая нормальная форма (5FN)
- 8.5.2. Правила Кодда
  - 8.5.2.1. Правило 1: Информационное правило
  - 8.5.2.2. Правило 2: Гарантированный доступ к данным
  - 8.5.2.3. Правило 3: Систематическая поддержка отсутствующих значений
  - 8.5.2.4. Правило 4: Описание базы данных
  - 8.5.2.5. Правило 5: Полнота подмножества языка
  - 8.5.2.6. Правило 6: Изменение представлений
  - 8.5.2.7. Правило 7. Вставка и обновление
  - 8.5.2.8. Правило 8: Физическая независимость данных
  - 8.5.2.9. Правило 9: Логическая независимость данных
  - 8.5.2.10. Правило 10: Независимость контроля целостности
    - 8.5.2.10.1. Правила целостности
  - 8.5.2.11. Правило 11: Независимость от расположения
  - 8.5.2.12. Правило 12: Согласование языковых уровней
- 8.5.3. Наглядный пример

8.6. Хранилище данных/OLAP-система
- 8.6.1. Хранилище данных
- 8.6.2. Таблица фактов
- 8.6.3. Таблица размеров
- 8.6.4. Создание системы OLAP. Инструменты

8.7. Производительность базы данных (БД)
- 8.7.1. Оптимизация индекса
- 8.7.2. Оптимизация запросов
- 8.7.3. Разбиение таблиц

8.8. Моделирование реального проекта для проектирования БД (I)
- 8.8.1. Обзор проекта (Компания A)
- 8.8.2. Применение проектирования баз данных
- 8.8.3. Предлагаемые тесты
- 8.8.4. Предлагаемые тесты. *Обратная связь*

8.9. Моделирование реального проекта для проектирования БД (II)
- 8.9.1. Обзор проекта (Компания B)
- 8.9.2. Применение проектирования баз данных
- 8.9.3. Предлагаемые упражнения
- 8.9.4. Предлагаемые упражнения. *Обратная связь*

8.10. Актуальность оптимизации БД для качества программного обеспечения
- 8.10.1. Оптимизация проектирования
- 8.10.2. Оптимизация кода запросов
- 8.10.3. Оптимизация кода хранимой процедуры
- 8.10.4. Влияние *триггеров* на качество программного обеспечения. Рекомендации по применению

Модуль 9. Проектирование масштабируемых архитектур. Архитектура в жизненном цикле программного обеспечения

9.1. Проектирование масштабируемых архитектур (I)
- 9.1.1. Масштабируемые архитектуры
- 9.1.2. Принципы масштабируемой архитектуры
  - 9.1.2.1. Надежность
  - 9.1.2.2. Масштабируемость
  - 9.1.2.3. Поддержка
- 9.1.3. Типы масштабируемости
  - 9.1.3.1. Вертикальная
  - 9.1.3.2. Горизонтальная
  - 9.1.3.3. Комбинированная

9.2. Предметно-ориентированное проектирование (*Domain-Driven Design*)
- 9.2.1. Модель DDD. Ориентация на домен
- 9.2.2. Слои, распределение ответственности и шаблоны проектирования
- 9.2.3. Разделение как основа для качества

9.3. Проектирование масштабируемых архитектур (II). Преимущества, ограничения и стратегии проектирования
- 9.3.1. Масштабируемая архитектура. Преимущества
- 9.3.2. Масштабируемая архитектура. Ограничения
- 9.3.3. Стратегии разработки масштабируемых архитектур (описательная таблица)

9.4. Жизненный цикл программного обеспечения (I). Этапы
- 9.4.1. Жизненный цикл программного обеспечения
  - 9.4.1.1. Этап планирования
  - 9.4.1.2. Этап анализа
  - 9.4.1.3. Этап проектирования
  - 9.4.1.4. Этап реализации
  - 9.4.1.5. Этап тестирования
  - 9.4.1.6. Этап установки/развертывания
  - 9.4.1.7. Этап использования и обслуживания

9.5. Модели жизненного цикла программного обеспечения
- 9.5.1. Каскадная модель
- 9.5.2. Повторяющаяся модель
- 9.5.3. Спиральная модель
- 9.5.4. Модель Большого взрыва

9.6. Жизненный цикл программного обеспечения (II). Автоматизация
- 9.6.1. Жизненный цикл разработки программного обеспечения. Решения
  - 9.6.1.1. Непрерывная интеграция и разработка (CI/CD)
  - 9.6.1.2. Agile-методологии
  - 9.6.1.3. DevOps / Производственные операции
- 9.6.2. Будущие тенденции
- 9.6.3. Практические примеры

9.7. Архитектура программного обеспечения в жизненном цикле программного обеспечения
- 9.7.1. Преимущества
- 9.7.2. Ограничения
- 9.7.3. Инструменты

9.8. Моделирование реального проекта для проектирования архитектуры программного обеспечения (I)
- 9.8.1. Обзор проекта (Компания A)
- 9.8.2. Применение проектирования архитектуры программного обеспечения
- 9.8.3. Предлагаемые тесты
- 9.8.4. Предлагаемые тесты. *Обратная связь*

9.9. Моделирование реального проекта Для для проектирования архитектуры программного обеспечения (II)
- 9.9.1. Обзор проекта (Компания B)
- 9.9.2. Применение проектирования архитектуры программного обеспечения
- 9.9.3. Предлагаемые тесты
- 9.9.4. Предлагаемые упражнения *Обратная связь*

9.10. Симуляция реального проекта для проектирования архитектуры программного обеспечения (III)
- 9.10.1. Обзор проекта (Компания C)
- 9.10.2. Применение проектирования архитектуры программного обеспечения
- 9.10.3. Предлагаемые упражнения
- 9.10.4. Предлагаемые упражнения. *Обратная связь*

Модуль 10. Критерии качества ISO, IEC 9126. Метрики качества программного обеспечения

10.1. Критерии качества. Стандарт ISO, IEC 9126
- 10.1.1. Критерии качества
- 10.1.2. Качество программного обеспечения. Обоснование. Стандарт ISO, IEC 9126
- 10.1.3. Измерение качества программного обеспечения как ключевой показатель

10.2. Критерии качества программного обеспечения. Характеристики
- 10.2.1. Надежность
- 10.2.2. Функциональность
- 10.2.3. Эффективность
- 10.2.4. Юзабилити
- 10.2.5. Обслуживаемость
- 10.2.6. Портативность

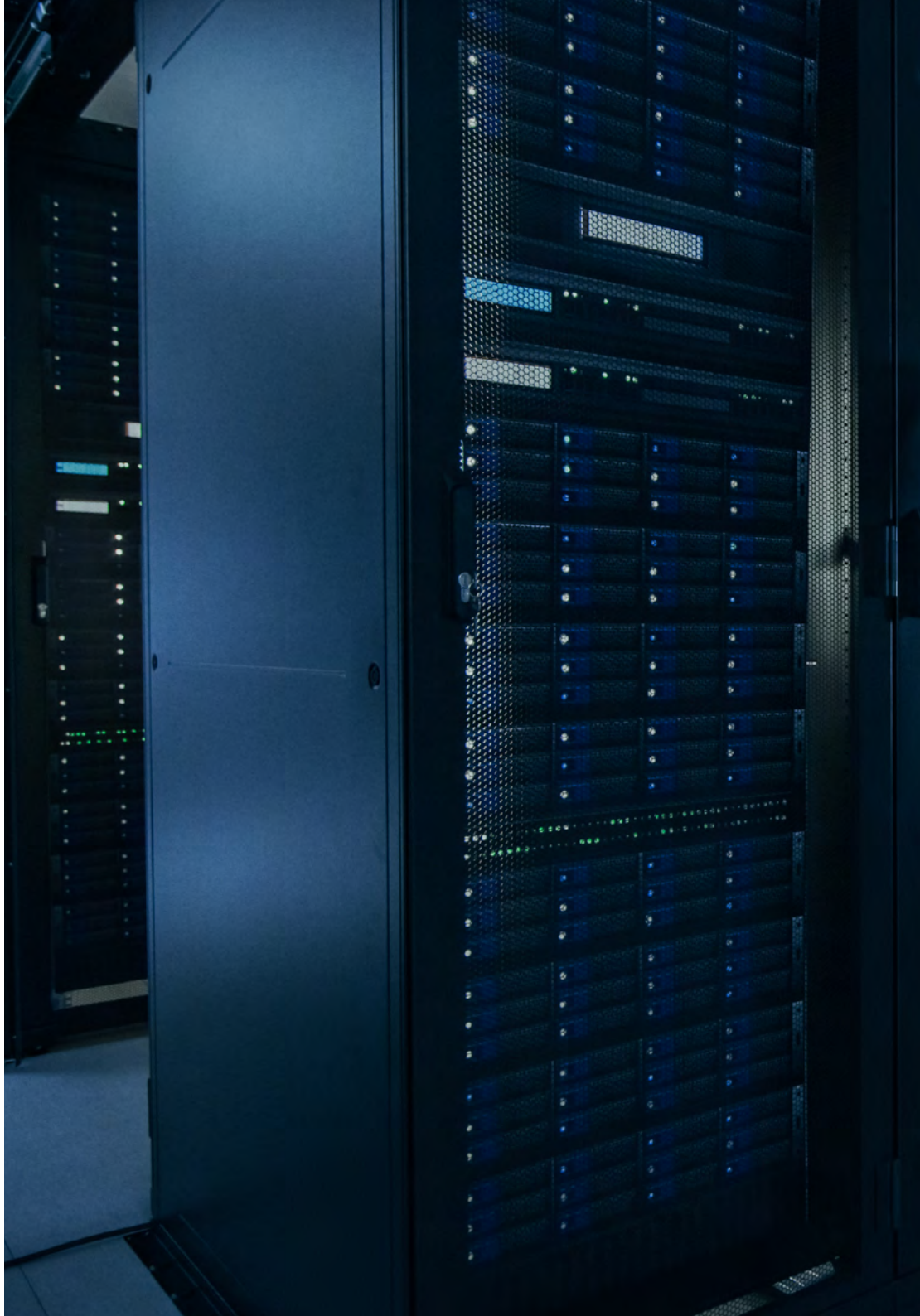
10.3. Стандарт ISO, IEC 9126 (I). Презентация
- 10.3.1. Описание нарушений Стандарт ISO, IEC 9126
- 10.3.2. Функциональность
- 10.3.3. Надежность
- 10.3.4. Юзабилити
- 10.3.5. Обслуживаемость
- 10.3.6. Портативность
- 10.3.7. Качество использования
- 10.3.8. Метрики качества программного обеспечения
- 10.3.9. Метрические данные качества в ISO 9126

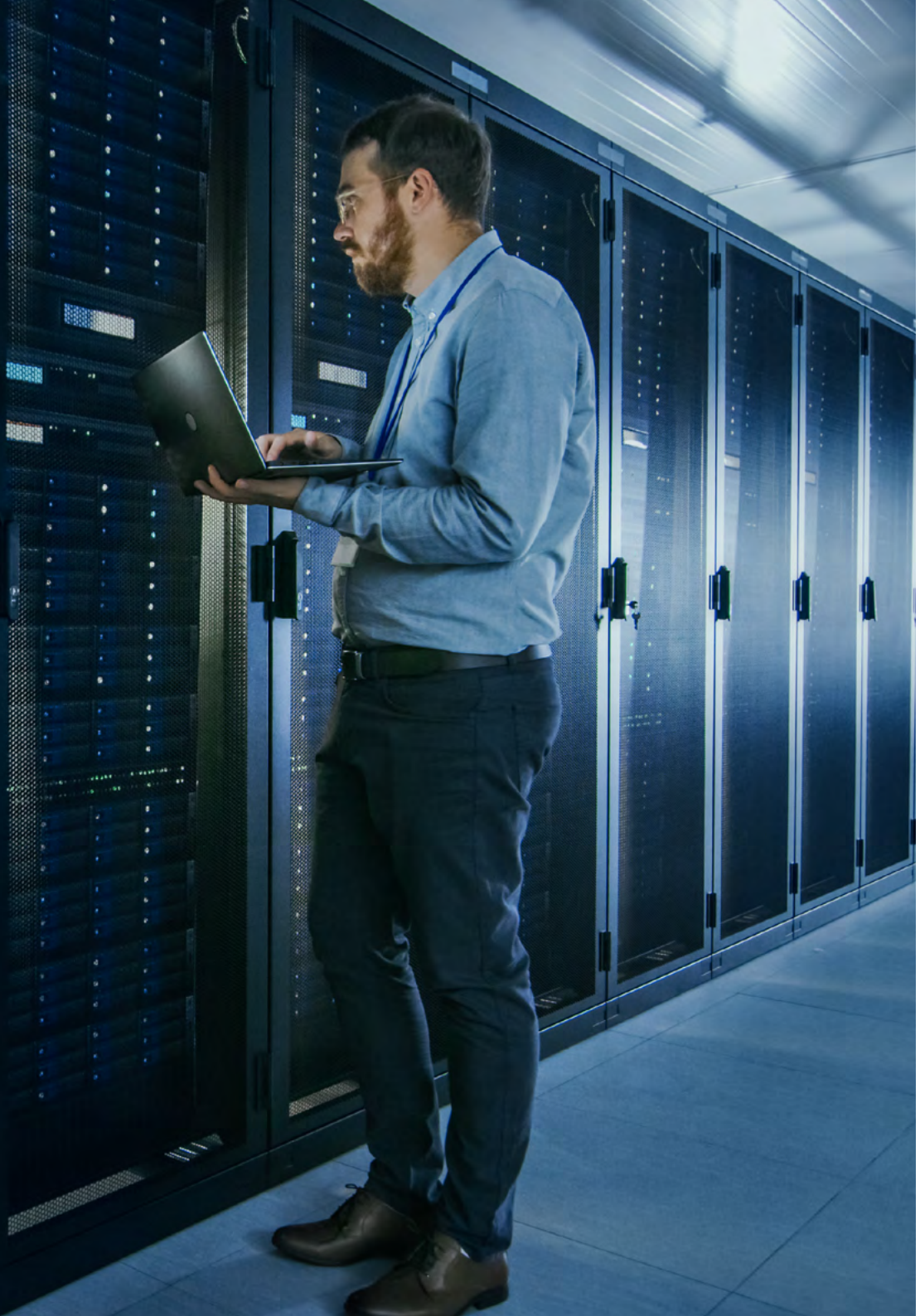
10.4. Стандарт ISO, IEC 9126 (II). Модели МакКолла и Боэма
- 10.4.1. Модель МакКолла: Факторы качества
- 10.4.2. Модель Боэма
- 10.4.3. Промежуточный уровень. Характеристики

10.5. Метрики качества программного обеспечения (I). Элементы
- 10.5.1. Измерения
- 10.5.2. Метрические
- 10.5.3. Показатели
  - 10.5.3.1. Типы показателей
- 10.5.4. Размеры и модели
- 10.5.5. Сфера применения метрики программного обеспечения
- 10.5.6. Классификация метрик программного обеспечения

10.6. Измерение качества программного обеспечения (II). Практика измерений
- 10.6.1. Сбор метрических данных
- 10.6.2. Измерение внутренних атрибутов продукта
- 10.6.3. Измерение внешних атрибутов продукта
- 10.6.4. Измерение ресурсов
- 10.6.5. Метрики для объектно-ориентированных систем

10.7. Разработка единого показателя качества программного обеспечения
- 10.7.1. Отдельный показатель как глобальный показатель
- 10.7.2. Разработка, обоснование и применение индикаторов
- 10.7.3. Примеры применения. Необходимость знать детали





10.8. Моделирование реального проекта для измерения качества (I)
- 10.8.1. Обзор проекта (Компания A)
- 10.8.2. Применение измерения качества
- 10.8.3. Предлагаемые тесты
- 10.8.4. Предлагаемые тесты. *Обратная связь*

10.9. Моделирование реального проекта для измерения качества (II)
- 10.9.1. Обзор проекта (Компания B)
- 10.9.2. Применение измерения качества
- 10.9.3. Предлагаемые тесты
- 10.9.4. Предлагаемые тесты. *Обратная связь*

10.10. Моделирование реального проекта для измерения качества (III)
- 10.10.1. Обзор проекта (Компания C)
- 10.10.2. Применение измерения качества
- 10.10.3. Предлагаемые тесты
- 10.10.4. Предлагаемые тесты. *Обратная связь*

Модуль 11. Методологии, разработка и качество в программировании

11.1. Разработка программного обеспечения на основе моделей
- 11.1.1. Необходимость
- 11.1.3. Моделирование объектов
- 11.1.4. UML
- 11.1.5. CASE-инструменты

11.2. Моделирование приложений и шаблоны проектирования с помощью UML
- 11.2.1. Расширенное моделирование требований
- 11.2.2. Усовершенствованное статическое моделирование
- 11.2.3. Продвинутое динамическое моделирование
- 11.2.4. Моделирование компонентов
- 11.2.5. Введение в шаблоны проектирования с UML
- 11.2.6. Adapter
- 11.2.7. Factory
- 11.2.8. Singleton
- 11.2.9. Strategy
- 11.2.10. Composite
- 11.2.11. Facade
- 11.2.12. Observer

11.3. Инженерия, управляемая моделями
- 11.3.1. Введение
- 11.3.2. Метамоделирование систем
- 11.3.3. MDA
- 11.3.4. DSL
- 11.3.5. Уточнения модели с помощью OCL
- 11.3.6. Преобразования моделей

11.4. Онтологии в программной инженерии
- 11.4.1. Введение
- 11.4.2. Инженерия онтологии
- 11.4.3. Применение онтологий в программной инженерии

Модуль 12. Управление программного обеспечения проекта

12.1. Управление *заинтересованными сторонами* и охватом
- 12.1.1. Выявление заинтересованных сторон
- 12.1.2. Разработка плана по управлению заинтересованными сторонами
- 12.1.3. Управление взаимодействием между заинтересованными сторонами
- 12.1.4. Контроль взаимодействия между заинтересованными сторонами
- 12.1.5. Цель проекта
- 12.1.6. Управление охватом и его план
- 12.1.7. Сбор информации о требованиях
- 12.1.8. Определение сферы применения
- 12.1.9. Создание WBS (EDT)
- 12.1.10. Утверждение и контроль масштаба

12.2. Разработка расписания
- 12.2.1. Управление временем и его планирование
- 12.2.2. Определение деятельности
- 12.2.3. Составление последовательности деятельности
- 12.2.4. Оценка ресурсов деятельности
- 12.2.5. Предполагаемая продолжительность деятельности
- 12.2.6. Разработка графика и расчет критического пути
- 12.2.7. Контроль расписания

12.3. Составление бюджета и реагирование на риски
- 12.3.1. Оценка затрат
- 12.3.2. Разработка бюджета и S-образной кривой
- 12.3.3. Контроль затрат и метод оценки стоимости
- 12.3.4. Понятие риска
- 12.3.5. Как проводить анализ рисков
- 12.3.6. Разработка плана реагирования

12.4. Коммуникация и человеческие ресурсы
- 12.4.1. Планирование управления коммуникациями
- 12.4.2. Анализ требований к коммуникациям
- 12.4.3. Коммуникационные технологии
- 12.4.4. Модели коммуникации
- 12.4.5. Методы коммуникации
- 12.4.6. План управления коммуникациями
- 12.4.7. Управление коммуникациями
- 12.4.8. Управление персоналом
- 12.4.9. Основные участники и их роли в проектах
- 12.4.10. Типы организаций
- 12.4.11. Организация проекта
- 12.4.12. Рабочая группа

12.5. Закупки
- 12.5.1. Процесс закупок
- 12.5.2. Планирование
- 12.5.3. Поиск поставщиков и запрос на тендеры
- 12.5.4. Заключение контракта
- 12.5.5. Администрирование контракта
- 12.5.6. Контракты
- 12.5.7. Виды контрактов
- 12.5.8. Ведение переговоров по контракту

12.6. Выполнение, мониторинг и контроль и закрытие
- 12.6.1. Группы процессов
- 12.6.2. Осуществление проекта
- 12.6.3. Наблюдение и контроль проекта
- 12.6.4. Закрытие проекта

12.7. Профессиональная ответственность
- 12.7.1. Профессиональная ответственность
- 12.7.2. Характеристики социальной и профессиональной ответственности
- 12.7.3. Кодекс этических норм руководителя проекта
- 12.7.4. Ответственность vs. PMP®
- 12.7.5. Примеры ответственности
- 12.7.6. Преимущества профессионализации

Модуль 13. Платформы для разработки программного обеспечения

13.1. Введение в разработку приложений
- 13.1.1. Приложения для настольных ПК
- 13.1.2. Язык программирования
- 13.1.3. Интегрированные среды разработки
- 13.1.4. Веб-приложения
- 13.1.5. Мобильные приложения
- 13.1.6. Облачные приложения

13.2. Разработка приложений и графического интерфейса на Java
- 13.2.1. Интегрированные среды разработки для Java
- 13.2.2. Основные IDE для Java
- 13.2.3. Знакомство с платформой разработки Eclipse
- 13.2.4. Знакомство с платформой разработки NetBeans
- 13.2.5. Модель View Controller для графических пользовательских интерфейсов
- 13.2.6. Создание графического интерфейса в Eclipse
- 13.2.7. Дизайн графического интерфейса в NetBeans

13.3. Отладка и тестирование в языке Java
- 13.3.1. Тестирование и отладка Java-программ
- 13.3.2. Отладка в Eclipse
- 13.3.3. Отладка в NetBeans

13.4. Разработка приложений и графического интерфейса на .NET
- 13.4.1. Net Framework
- 13.4.2. Компоненты платформы разработки .NET
- 13.4.3. Визуальная студия .NET
- 13.4.4. Инструменты графического интерфейса .NET
- 13.4.5. Графический интерфейс с Windows Presentation Foundation
- 13.4.6. Отладка и компиляция приложения WPF

13.5. Программирование для сетей .NET
- 13.5.1. Введение в сетевое программирование в .NET
- 13.5.2. Запросы и ответы в .NET
- 13.5.3. Использование прикладных протоколов в .NET
- 13.5.4. Безопасность при программировании сетей .NET

13.6. Среды разработки мобильных приложений
- 13.6.1. Мобильные приложения
- 13.6.2. Мобильные приложения Android
- 13.6.3. Шаги для разработки Android
- 13.6.4. Интегрированная среда разработки Android Studio

13,7. Разработка приложений в среде Android Studio
- 13.7.1. Установить и запустить Android Studio
- 13.7.2. Запуск Android-приложения
- 13.7.3. Разработка графического интерфейса в Android Studio
- 13.7.4. Запуск действий в Android Studio

13.8. Отладка и публикация Android-приложений
- 13.8.1. Отладка приложения в Android Studio
- 13.8.2. Сохранение приложений в Android Studio
- 13.8.3. Публикация приложения в Google Play

13.9. Разработка приложений для облака
- 13.9.1. *Облачные вычисления*
- 13.9.2. Уровни *cloud*: SaaS, PaaS, IaaS
- 13.9.3. Основные платформы для разработки облаков
- 13.9.4. Библиографические ссылки

13.10. Введение в Google Cloud Platform
- 13.10.1. Основы Google Cloud Platform
- 13.10.2. Услуги Google Cloud Platform
- 13.10.3. Инструменты Google Cloud Platform

Модуль 14. Вычисления на стороне клиента в веб-разработке

14.1. Введение в HTML
- 14.1.1. Структура документа
- 14.1.2. Цвет
- 14.1.3. Текст
- 14.1.4. Гипертекстовые ссылки
- 14.1.5. Изображения
- 14.1.6. Списки
- 14.1.7. Таблицы
- 14.1.8. Рамки (*frames*)
- 14.1.9. Формуляры
- 14.1.10. Специфические элементы для мобильных технологий
- 14.1.11. Неиспользуемые предметы

14.2. Таблицы веб-стиля (CSS)
- 14.2.1. Элементы и структура таблицы стилей
  - 14.2.1.1. Создание таблиц стилей
  - 14.2.1.2. Применение стилей. Селекторы
  - 14.2.1.3. Наследование стилей и каскадирование
  - 14.2.1.4. Форматирование страниц с помощью стилей
  - 14.2.1.5. Форматирование страниц с помощью стилей. Модель коробки
- 14.2.2. Стили дизайна для различных устройств
- 14.2.3. Типы таблиц стилей: статические и динамические. Псевдоклассы
- 14.2.4. Передовая практика использования таблиц стилей

14.3. Введение и история JavaScript
- 14.3.1. Введение
- 14.3.2. История JavaScript
- 14.3.3. Среда разработки для использования

14.4. Основные понятия веб-программирования
- 14.4.1. Основной синтаксис JavaScript
- 14.4.2. Примитивные типы данных и операторов
- 14.4.3. Переменные и области
- 14.4.4. Текстовые строки и *шаблонные литералы*
- 14.4.5. Числа и логические значения
- 14.4.6. Сравнения

14.5. Сложные структуры JavaScript
- 14.5.1. Векторы или *массивы* и объекты
- 14.5.2. Множества
- 14.5.3. Карты
- 14.5.4. Дизъюнкция
- 14.5.5. Циклы

14.6. Функции и объекты
- 14.6.1. Определение и вызов функций
- 14.6.2. Аргументы
- 14.6.3. Функции стрелок
- 14.6.4. Функции обратной связи или *callback*
- 14.6.5. Функции высшего порядка
- 14.6.6. Буквальные объекты
- 14.6.7. Объект *this*
- 14.6.8. Объекты как пространства имен: объект Math и объект Date

14.7. Объектная модель документа (DOM)
- 14.7.1. Что такое DOM?
- 14.7.2. Немного истории
- 14.7.3. Навигация и получение элементов
- 14.7.4. Виртуальный DOM с помощью JSDOM
- 14.7.5. Селекторы запросов или *query selectors*
- 14.7.6. Навигация по свойствам
- 14.7.7. Распределение элементов по атрибутам
- 14.7.8. Создание и изменение узлов
- 14.7.9. Обновление стиля элементов DOM

14.8. Современная веб-разработка
- 14.8.1. Поток, управляемый событиями, и *слушатели*
- 14.8.2. Современные веб *toolkits* и системы выравнивания
- 14.8.3. Строгий режим JavaScript
- 14.8.4. Более подробно о функциях
- 14.8.5. Асинхронные обещания и функции
- 14.8.6. *Закрытия*
- 14.8.7. Функциональное программирование
- 14.8.8. POO в JavaScript

14.9. Веб-юзабилити
- 14.9.1. Введение в удобство использования
- 14.9.2. Определение удобства использования
- 14.9.3. Методология дизайна, ориентированного на пользователя
- 14.9.4. Различия между доступностью и удобством использования
- 14.9.5. Преимущества и проблемы сочетания доступности и удобства использования
- 14.9.6. Преимущества и трудности при внедрении веб-сайтов
- 14.9.7. Методы удобства использования
- 14.9.8. Анализ требований пользователей
- 14.9.9. Принципы концептуального дизайна. Прототипирование, ориентированное на пользователя
- 14.9.10. Руководство по созданию веб-сайтов
  - 14.9.10.1. Рекомендации по использованию Jakob Nielsen
  - 14.9.10.2. Рекомендации по использованию Bruce Tognazzini
- 14.9.11. Оценка юзабилити

14.10. Доступность веб-сайтов
- 14.10.1. Введение
- 14.10.2. Определение веб-доступности
- 14.10.3. Виды инвалидности
  - 14.10.3.1. Временная или постоянная инвалидность
  - 14.10.3.2. Нарушение зрения
  - 14.10.3.3. Нарушение слуха
  - 14.10.3.4. Двигательные нарушения
  - 14.10.3.5. Неврологические или когнитивные нарушения
  - 14.10.3.6. Трудности, связанные со старением
  - 14.10.3.7. Ограничения, возникающие в связи с окружающей средой
  - 14.10.3.8. Барьеры, препятствующие доступу в интернет
- 14.10.4. Технические средства и вспомогательные продукты для преодоления барьеров
  - 14.10.4.1. Помощь для слепых людей
  - 14.10.4.2. Помощь для людей со слабым зрением
  - 14.10.4.3. Помощь для людей с дальтонизмом
  - 14.10.4.4. Помощь для людей с ограниченными возможностями слуха
  - 14.10.4.5. Помощь для людей с ограниченными двигательными возможностями
  - 14.10.4.6. Помощь для людей с когнитивными и неврологическими нарушениями
- 14.10.5. Преимущества и трудности при внедрении полезных веб-сайтов
- 14.10.6. Правила и стандарты веб-доступности
- 14.10.7. Органы регулирования веб-доступности
- 14.10.8. Сравнение норм и стандартов
- 14.10.9. Руководство по соблюдению нормативных актов и стандартов
  - 14.10.9.1. Описание основных руководящих принципов (изображения, ссылки, видео и т.д.)
  - 14.10.9.2. Руководство по доступной навигации
    - 14.10.9.2.1. Восприимчивость
    - 14.10.9.2.2. Работоспособность
    - 14.10.9.2.3. Понятность
    - 14.10.9.2.4. Устойчивость
- 14.10.10. Описание процесса обеспечения соответствия требованиям веб-доступности
- 14.10.11. Уровни соответствия
- 14.10.12. Критерии соответствия
- 14.10.13. Требования к соответствию
- 14.10.14. Методология оценки доступности веб-сайтов

Модуль 15. Вычисления на стороне веб-сервера

15.1. Введение в программирование сервера: PHP
- 15.1.1. Основы серверного программирования
- 15.1.2. Основной синтаксис PHP
- 15.1.3. Генерация HTML-контента с помощью PHP
- 15.1.4. Среды разработки и тестирования: XAMPP

15.2. Продвинутый PHP
- 15.2.1. Структуры управления в PHP
- 15.2.2. Функции в PHP
- 15.2.3. Работа с *массивами* в PHP
- 15.2.4. Работа со строками в PHP
- 15.2.5. Объектно-ориентированный PHP

15.3. Модели данных
- 15.3.1. Понятие данных. Жизненный цикл данных
- 15.3.2. Типы данных
  - 15.3.2.1. Основные
  - 15.3.2.2. Записи
  - 15.3.2.3. Динамические

15.4. Реляционная модель
- 15.4.1. Описание
- 15.4.2. Организации и типы организаций
- 15.4.3. Элементы данных. Атрибуты
- 15.4.4. Отношения: типы, подтипы, кардинальность
- 15.4.5. Пароли. Типы паролей
- 15.4.6. Нормализация. Нормальные формы

15.5. Построение логической модели данных
- 15.5.1. Спецификация таблиц
- 15.5.2. Определение столбцов
- 15.5.3. Спецификация паролей
- 15.5.4. Переход к нормальным формам. Зависимости

15.6. Физическая модель данных. Файлы данных
- 15.6.1. Описание файлов данных
- 15.6.2. Типы файлов
- 15.6.3. Режимы доступа
- 15.6.4. Организация файлов

15.7. Доступ к базе данных из PHP
- 15.7.1. Введение в MariaDB
- 15.7.2. Работа с базой данных MariaDB: язык SQL
- 15.7.3. Доступ к базе данных из PHP
- 15.7.4. Введение в MySql
- 15.7.5. Работа с базой данных MySql: язык SQL
- 15.7.6. Доступ к базе данных MySql из PHP

15.8. Взаимодействие с клиентом из PHP
- 15.8.1. Формы PHP
- 15.8.2. Файлы cookies
- 15.8.3. Управление сессиями

15.9. Архитектура веб-приложений
- 15.9.1. Схема Модель-Представление-Контроллер
- 15.9.2. Контроллер
- 15.9.3. Модель
- 15.9.4. Представление

15,10. Введение в веб-сервисы
- 15.10.1. Введение в XML
- 15.10.2. Сервис-ориентированные архитектуры (SOA): веб-сервисы
- 15.10.3. Создание веб-сервисов SOAP и REST
- 15.10.4. Протокол SOAP
- 15.10.5. Протокол REST

Модуль 16. Управление безопасностью

16.1. Визуализация информации
- 16.1.1. Введение
- 16.1.2. Информационная безопасность подразумевает конфиденциальность, целостность и доступность
- 16.1.3. Безопасность - это экономический вопрос
- 16.1.4. Безопасность - это процесс
- 16.1.5. Классификация информации
- 16.1.6. Информационная безопасность включает в себя управление рисками
- 16.1.7. Безопасность связана с элементами управления безопасностью
- 16.1.8. Безопасность является как физической, так и логической
- 16.1.9. Безопасность включает в себя людей

16.2. Специалист по информационной безопасности
- 16.2.1. Введение
- 16.2.2. Информационная безопасность как профессия
- 16.2.3. Сертификация (ISC) 2
- 16.2.4. Стандарт ISO 27001
- 16.2.5. Эффективные методы обеспечения безопасности при управлении ИТ-услугами
- 16.2.6. Модели зрелости для информационной безопасности
- 16.2.7. Другие сертификаты, стандарты и профессиональные ресурсы

16.3. Контроль доступа
- 16.3.1. Введение
- 16.3.2. Требования по контролю доступа
- 16.3.3. Механизмы аутентификации
- 16.3.4. Методы авторизации
- 16.3.5. Учет доступа и аудит
- 16.3.6. Технологии тройного А

16.4. Программы, процессы и политики информационной безопасности
- 16.4.1. Введение
- 16.4.2. Программы управления безопасностью
- 16.4.3. Управление рисками
- 16.4.4. Разработка политики безопасности

16,5. Планы обеспечения непрерывности бизнеса (BCP)
- 16.5.1. Введение в BCP
- 16.5.2. Стадии I и II
- 16.5.3. Стадии III и IV
- 16.5.4. Обслуживание BCP

16.6. Процедуры для надлежащей защиты компании
- 16.6.1. Сети DMZ
- 16.6.2. Системы обнаружения вторжений
- 16.6.3. Требования по контролю доступа
- 16.6.4. Обучение у нападающего: Honeypot

16.7. Архитектура безопасности. Профилактика
- 16.7.1. Обзор. Деятельность и многоуровневая модель
- 16.7.2. Защита периметра (межсетевые экраны, WAF, IPS и т.д.)
- 16.7.3. Защита конечных точек (оборудование, серверы и услуги)

16.8. Архитектура безопасности. Обнаружение
- 16.8.1. Обнаружение и мониторинг обзора
- 16.8.2. Журналы, зашифрованное разбиение трафика, запись и Siems
- 16.8.3. Оповещения и разведка

16.9. Архитектура безопасности. Реакция
- 16.9.1. Реакция. Продукты, услуги и ресурсы
- 16.9.2. Управление инцидентами
- 16.9.3. CERTS и CSIRT

16.10. Архитектура безопасности. Восстановление
- 16.10.1. Устойчивость, концепции, бизнес-требования и стандарты
- 16.10.2. Устойчивость ИТ-решений
- 16.10.3. Антикризисное управление и руководство

Модуль 17. Безопасность программного обеспечения

17.1. Вопросы безопасности программного обеспечения
- 17.1.1. Введение в проблему безопасности программного обеспечения
- 17.1.2. Уязвимости и их классификация
- 17.1.3. Свойства безопасного программного обеспечения
- 17.1.4. Рекомендации

17.2. Принципы проектирования безопасности программного обеспечения
- 17.2.1. Введение
- 17.2.2. Принципы проектирования безопасности программного обеспечения
- 17.2.3. Типы S-SDLC
- 17.2.4. Безопасность программного обеспечения на этапах S-SDLC
- 17.2.5. Методологии и стандарты
- 17.2.6. Рекомендации

17.3. Безопасность в жизненном цикле программного обеспечения на этапах разработки требований и проектирования
- 17.3.1. Введение
- 17.3.2. Моделирование атак
- 17.3.3. Случаи жестокого обращения
- 17.3.4. Разработка требований безопасности
- 17.3.5. Анализ риска. Архитектура
- 17.3.6. Модели проектирования
- 17.3.7. Рекомендации

17.4. Безопасность жизненного цикла программного обеспечения на этапах кодирования, тестирования и эксплуатации
- 17.4.1. Введение
- 17.4.2. Тестирование безопасности с учетом рисков
- 17.4.3. Обзор кода
- 17.4.4. Тест на проникновение
- 17.4.5. Операции по обеспечению безопасности
- 17.4.6. Внешний обзор
- 17.4.7. Рекомендации

17.5. Приложения для безопасного кодирования I
- 17.5.1. Введение
- 17.5.2. Практика безопасного кодирования
- 17.5.3. Обработка и проверка записей
- 17.5.4. Переполнение памяти
- 17.5.5. Рекомендации

17.6. Приложения для безопасного кодирования II
- 17.6.1. Введение
- 17.6.2. Переполнения целых чисел, ошибки усечения и проблемы с преобразованием типов между целыми числами
- 17.6.3. Ошибки и исключения
- 17.6.4. Приватность и конфиденциальность
- 17.6.5. Привилегированные программы
- 17.6.6. Рекомендации

17.7. Безопасность в разработке и в облаке
- 17.7.1. Безопасность в развитии; методология и практика
- 17.7.2. Модели PaaS, IaaS, CaaS и SaaS
- 17.7.3. Безопасность в облаке и для облачных услуг

17.8. Шифрование
- 17.8.1. Основы криптологии
- 17.8.2. Симметричное и асимметричное шифрование
- 17.8.3. Шифрование в состоянии покоя и при транспортировке

17.9. Оркестровка и автоматизация безопасности (SOAR)
- 17.9.1. Сложность ручной обработки; необходимость автоматизации задач
- 17.9.2. Продукты и услуги
- 17.9.3. Архитектура SOAR

17.10. Безопасность при удаленной работе
- 17.10.1. Потребность и сценарии
- 17.10.2. Продукты и услуги
- 17.10.3. Безопасность при удаленной работе

Модуль 18. Администрирование веб-сервера

18.1. Введение в веб-серверы
- 18.1.1. Что такое веб-сервер?
- 18.1.2. Архитектура и работа веб-сервера
- 18.1.3. Ресурсы и содержание веб-сервера
- 18.1.4. Серверы приложений
- 18.1.5. Прокси-серверы
- 18.1.6. Основные веб-серверы на рынке
- 18.1.7. Статистика использования веб-сервера
- 18.1.8. Безопасность веб-сервера
- 18.1.9. Балансировка нагрузки на веб-серверах
- 18.1.10. Рекомендации

18.2. Работа с протоколом HTTP
- 18.2.1. Функционирование и структура.
- 18.2.2. Описание запросов или методов запросов
- 18.2.3. Коды состояния
- 18.2.4. Заголовки
- 18.2.5. Кодирование содержимого. Кодовые страницы
- 18.2.6. Выполнение HTTP-запросов в Интернете с использованием прокси, livehttpheaders или аналогичного метода, анализ используемого протокола

18.3. Описание многосерверных распределенных архитектур
- 18.3.1. 3-слойная модель
- 18.3.2. Устойчивость к сбоям
- 18.3.3. Распределение нагрузки
- 18.3.4. Хранилища состояния сеанса
- 18.3.5. Хранилища кэша

18.4. Информационные службы Интернета (IIS)
- 18.4.1. Что такое IIS?
- 18.4.2. История и эволюция IIS
- 18.4.3. Ключевые преимущества и особенности IIS7 и последующих версий
- 18.4.4. Архитектура IIS7 и последующих версий

18.5. Установка, администрирование и настройка IIS
- 18.5.1. Преамбула
- 18.5.2. Установка информационных служб Интернета (IIS)
- 18.5.3. Инструменты администрирования IIS
- 18.5.4. Создание, настройка и администрирование веб-сайтов
- 18.5.5. Установка и обработка расширений в IIS

18.6. Расширенная безопасность в IIS
- 18.6.1. Преамбула
- 18.6.2. Аутентификация, авторизация и контроль доступа в IIS
- 18.6.3. Настройка защищенного веб-сайта на IIS с помощью SSL
- 18.6.4. Политики безопасности, применяемые в IIS 8.x

18.7. Введение в Apache
- 18.7.1. Что такое Apache?
- 18.7.2. Основные преимущества Apache
- 18.7.3. Основные характеристики Apache
- 18.7.4. Архитектура

18.8. Установка и настройка Apache
- 18.8.1. Начальная установка Apache
- 18.8.2. Настройка Apache

18.9. Установка и настройка различных модулей в Apache
- 18.9.1. Установка модулей Apache
- 18.9.2. Типы модулей
- 18.9.3. Безопасная настройка Apache

18.10. Расширенная безопасность
- 18.10.1. Аутентификация, авторизация и контроль доступа
- 18.10.2. Методы аутентификации
- 18.10.3. Безопасная конфигурация Apache с помощью SSL

Модуль 19. Аудит безопасности

19.1. Введение в информационные системы и их аудит
- 19.1.1. Введение в информационные системы и роль компьютерного аудита
- 19.1.2. Определения ИТ-аудита и внутреннего ИТ-контроля
- 19.1.3. Функции и цели ИТ-аудита
- 19.1.4. Различия между внутренним контролем и ИТ-аудитом

19.2. Внутренний контроль информационных систем
- 19.2.1. Функциональная блок-схема центра обработки данных
- 19.2.2. Классификация средств контроля информационных систем
- 19.2.3. Золотое правило

19.3. Процесс и этапы аудита информационных систем
- 19.3.1. Оценка рисков (EDR) и другие методологии ИТ-аудита
- 19.3.2. Проведение аудита информационных систем. Этапы аудита
- 19.3.3. Ключевые навыки аудитора информационных систем

19.4. Технический аудит безопасности систем и сетей
- 19.4.1. Технические аудиты безопасности. Проверка на вторжение. Предварительные концепции
- 19.4.2. Аудит безопасности в системах. Инструменты поддержки
- 19.4.3. Аудит безопасности в сети. Инструменты поддержки

19.5. Технический аудит безопасности интернета и мобильных устройств
- 19.5.1. Аудит безопасности интернета. Инструменты поддержки
- 19.5.2. Аудит безопасности мобильных устройств. Инструменты поддержки
- 19.5.3. Приложение 1. Структура исполнительного отчета и технического отчета
- 19.5.4. Приложение 2. Набор инструментов
- 19.5.5. Приложение 3. Методики

19.6. Система управления информационной безопасностью
- 19.6.1. Безопасность ИТ: свойства и факторы влияния
- 19.6.2. Общеорганизационный риск и управление рисками: внедрение механизмов контроля
- 19.6.3. Система управления информационной безопасностью (СУИБ): концепция и критические факторы успеха
- 19.6.4. СУИБ - модель PDCA
- 19.6.5. СУИБ ISO-IEC 27001: организационный контекст
- 19.6.6. Организационная трансформация
- 19.6.7. Лидерство

19.6.8. Планирование
19.6.9. Поддержка
19.6.10. Эксплуатация
19.6.11. Оценка эффективности
19.6.12. Улучшение
19.6.13. Приложение к ISO 27001/ISO-IEC 27002: цели и средства контроля
19.6.14. Аудит ISMS
19.7. Проведение аудита
19.7.1. Процедуры
19.7.2. Техники
19.8. Прослеживаемость
19.8.1. Методики
19.8.2. Анализ
19.9. Хранение
19.9.1. Техники
19.9.2. Результаты
19.10. Отчетность и представление доказательств
19.10.1. Типы отчетов
19.10.2. Анализ данных
19.10.3. Представление доказательств

Модуль 20. Безопасность в онлайн-приложениях

20.1. Уязвимости и проблемы безопасности в онлайн-приложениях
20.1.1. Введение в безопасность в онлайн-приложениях
20.1.2. Уязвимости безопасности при разработке веб-приложений
20.1.3. Уязвимости безопасности при внедрении веб-приложений
20.1.4. Уязвимости безопасности при развертывания веб-приложений
20.1.5. Официальные списки уязвимостей безопасности
20.2. Политики и стандарты для обеспечения безопасности онлайн-приложений
20.2.1. Основные принципы обеспечения безопасности онлайн-приложений
20.2.2. Политика безопасности
20.2.3. Система управления информационной безопасностью
20.2.4. Жизненный цикл разработки безопасного программного обеспечения
20.2.5. Стандарты безопасности приложений

20.3. Безопасность при разработке веб-приложений
- 20.3.1. Введение в безопасность веб-приложений
- 20.3.2. Безопасность при разработке веб-приложений

20.4. Тестирование онлайн безопасности веб-приложений
- 20.4.1. Анализ и тестирование безопасности веб-приложений
- 20.4.2. Безопасность при развертывании и производстве веб-приложений

20.5. Безопасность веб-сервисов
- 20.5.1. Введение в безопасность веб-сервисов
- 20.5.2. Функции и технологии обеспечения безопасности веб-сервисов

20.6. Тестирование онлайн безопасности веб-приложений
- 20.6.1. Оценка безопасности веб-сервисов
- 20.6.2. Онлайн-защита. Межсетевые экраны и шлюзы XML

20.7. Этичный хакинг, вредоносное ПО и криминалистика
- 20.7.1. Этичный хакинг
- 20.7.2. Анализ вредоносных программ
- 20.7.3. Криминалистический анализ

20.8. Эффективные методы обеспечения безопасности приложений
- 20.8.1. Руководство по надлежащей практике разработки онлайн-приложений
- 20.8.2. Руководство по передовой практике внедрения онлайн-приложений

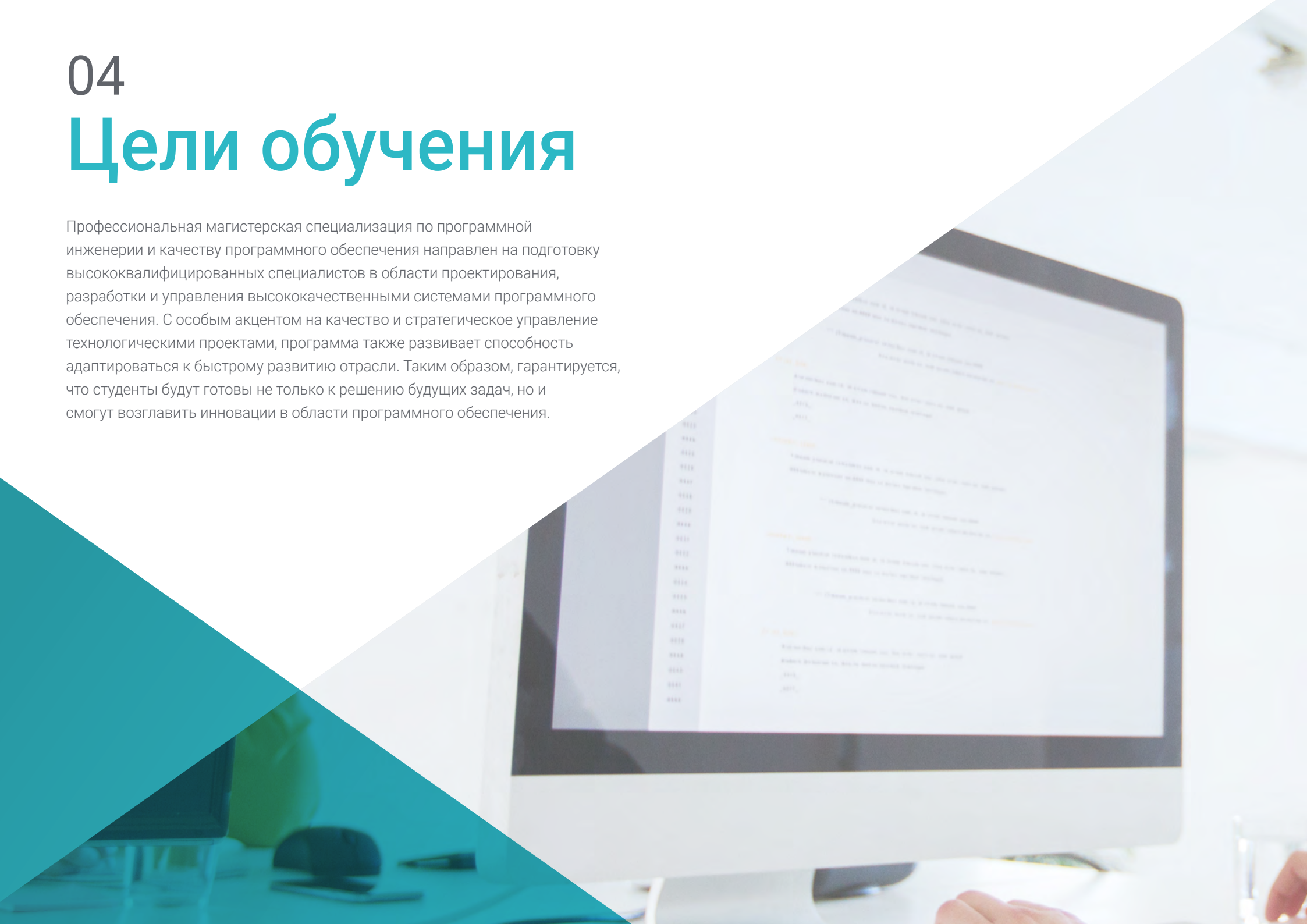
20.9. Распространенные ошибки, подрывающие безопасность приложений
- 20.9.1. Распространенные ошибки при разработке
- 20.9.2. Распространенные ошибки в хостинге
- 20.9.3. Распространенные ошибки в производстве

Комплексная учебная программа, которая позволит вам освоить область больших данных и стать успешным архитектором бизнес-стратегий"

04

Цели обучения

Профессиональная магистерская специализация по программной инженерии и качеству программного обеспечения направлен на подготовку высококвалифицированных специалистов в области проектирования, разработки и управления высококачественными системами программного обеспечения. С особым акцентом на качество и стратегическое управление технологическими проектами, программа также развивает способность адаптироваться к быстрому развитию отрасли. Таким образом, гарантируется, что студенты будут готовы не только к решению будущих задач, но и смогут возглавить инновации в области программного обеспечения.



Превращайте идеи в эффективные технологические решения с помощью искусства программной инженерии"

Общие цели

- Развить продвинутые навыки в проектировании, разработке и обслуживании сложных и масштабируемых систем программного обеспечения, применяя лучшие практики и методологии программной инженерии
- Обучить студентов обеспечению качества программного обеспечения, предоставив им инструменты и методы для гарантирования надежности, безопасности и производительности технологических решений
- Стимулировать лидерство в управлении технологическими проектами, развивая компетенции в руководстве многопрофильными командами, стратегическом планировании и принятии решений в динамичных условиях
- Продвигать способность адаптироваться к быстрым технологическим изменениям, специализируясь на новых инструментах, методах и трендах, которые позволяют оставаться на переднем крае инженерии программного обеспечения
- Развить компетенции в управлении качеством на всех этапах жизненного цикла программного обеспечения, от начального планирования до обслуживания и непрерывного улучшения систем
- Укрепить навыки коммуникации и работы в команде, необходимые для эффективного взаимодействия с различными заинтересованными сторонами, управления ожиданиями и обеспечения успеха технологических проектов

Совершенствуйте свои навыки и станьте лидером в создании передовых технологических решений"

Конкретные цели

Модуль 1. Качество программного обеспечения. Уровни развития TRL

- Понимать различные уровни технологической зрелости и их связь с качеством программного обеспечения
- Оценивать разработку программного обеспечения на каждом этапе TRL и ее влияние на конечное качество продукта

Модуль 2. Разработка программных проектов. Функциональная и техническая документация.

- Развивать навыки создания четкой и детализированной функциональной и технической документации в программных проектах
- Анализировать важность точной документации для управления проектами и качества программного обеспечения

Модуль 3. *Тестирование* программного обеспечения. Автоматизация тестирования

- Развивать навыки проектирования и выполнения автоматизированных тестов в программных приложениях
- Внедрять эффективные решения для *тестирования* с использованием инструментов автоматизации тестирования

Модуль 4. Методологии управления проектами программного обеспечения. Каскадная модель vs agile-методологии

- Анализировать различия между методологиями Waterfall и Agile в управлении проектами программного обеспечения
- Оценивать преимущества и ограничения каждой методологии в зависимости от типа проекта

Модуль 5. TDD (*Test Driven Developement*). Разработка программного обеспечения через тестирование

- Развивать навыки написания модульных тестов перед написанием кода для продакшна
- Улучшать качество программного обеспечения путем внедрения TDD в процесс разработки

Модуль 6. DevOps. Управление качеством программного обеспечения

- Исследовать концепцию DevOps и ее влияние на непрерывное повышение качества программного обеспечения
- Учиться интегрировать практики разработки и эксплуатации для достижения более гибкого и эффективного жизненного цикла программного обеспечения

Модуль 7. DevOps и непрерывная интеграция. Передовые практические решения в разработке программного обеспечения

- Глубже изучать передовые техники непрерывной интеграции в рамках DevOps
- Внедрять практические решения по непрерывной интеграции для автоматизации процесса разработки и развертывания программного обеспечения

Модуль 8. Проектирование баз данных (БД). Стандартизация и производительность. Качество программного обеспечения

- Анализировать принципы проектирования баз данных, включая нормализацию и оптимизацию производительности
- Понимать, как правильное проектирование баз данных способствует качеству программного обеспечения

Модуль 9. Проектирование масштабируемых архитектур. Архитектура в жизненном цикле программного обеспечения

- Глубже изучать принципы проектирования масштабируемых архитектур и их влияние на качество и производительность программного обеспечения
- Оценивать различные архитектурные шаблоны для создания масштабируемых программных приложений

Модуль 10. Критерии качества ISO, IEC 9126. Метрики качества программного обеспечения

- Понимать критерии качества программного обеспечения согласно данным стандартам и их применение
- Внедрять метрики качества для оценки и постоянного улучшения программных приложений

Модуль 11. Методологии, разработка и качество в программной инженерии

- Глубже изучать наиболее распространенные методологии в инженерии программного обеспечения и их связь с качеством
- Развивать комплексный подход, объединяющий разработку, тестирование и качество в проектах программного обеспечения

Модуль 12. Управление программного обеспечения проекта

- Развивать навыки управления проектами программного обеспечения — от планирования до реализации
- Управлять ресурсами, сроками и рисками, связанными с проектами разработки программного обеспечения

Модуль 13. Платформы для разработки программного обеспечения

- Исследовать различные платформы для разработки программного обеспечения и их характеристики
- Оценивать платформы разработки с точки зрения их возможностей, гибкости и совместимости с различными проектами

Модуль 14. Вычисления на стороне клиента в веб-разработке

- Анализировать, как выполняются вычисления на стороне клиента при разработке веб-приложений
- Разрабатывать приложения, использующие клиентские вычисления для улучшения взаимодействия и производительности

Модуль 15. Вычисления на стороне веб-сервера

- Исследовать технологии и методы, применяемые для вычислений на веб-сервере
- Понимать обработку данных, бизнес-логику и управление пользователями на сервере

Модуль 16. Управление безопасностью

- Оценивать риски безопасности в приложениях и применять превентивные меры
- Внедрять меры контроля безопасности на всех этапах жизненного цикла программного обеспечения

Модуль 17. Безопасность программного обеспечения

- Изучать лучшие практики безопасности при разработке программного обеспечения
- Анализировать наиболее распространенные уязвимости программного обеспечения и методы их устранения

Модуль 18. Администрирование веб-сервера

- Понимать роль веб-серверов в разработке и развертывании приложений
- Развивать навыки администрирования и поддержки веб-серверов

Модуль 19. Аудит безопасности

- Оценивать безопасность систем с помощью аудитов и тестирования на проникновение
- Внедрять процессы постоянного аудита для повышения безопасности программного обеспечения

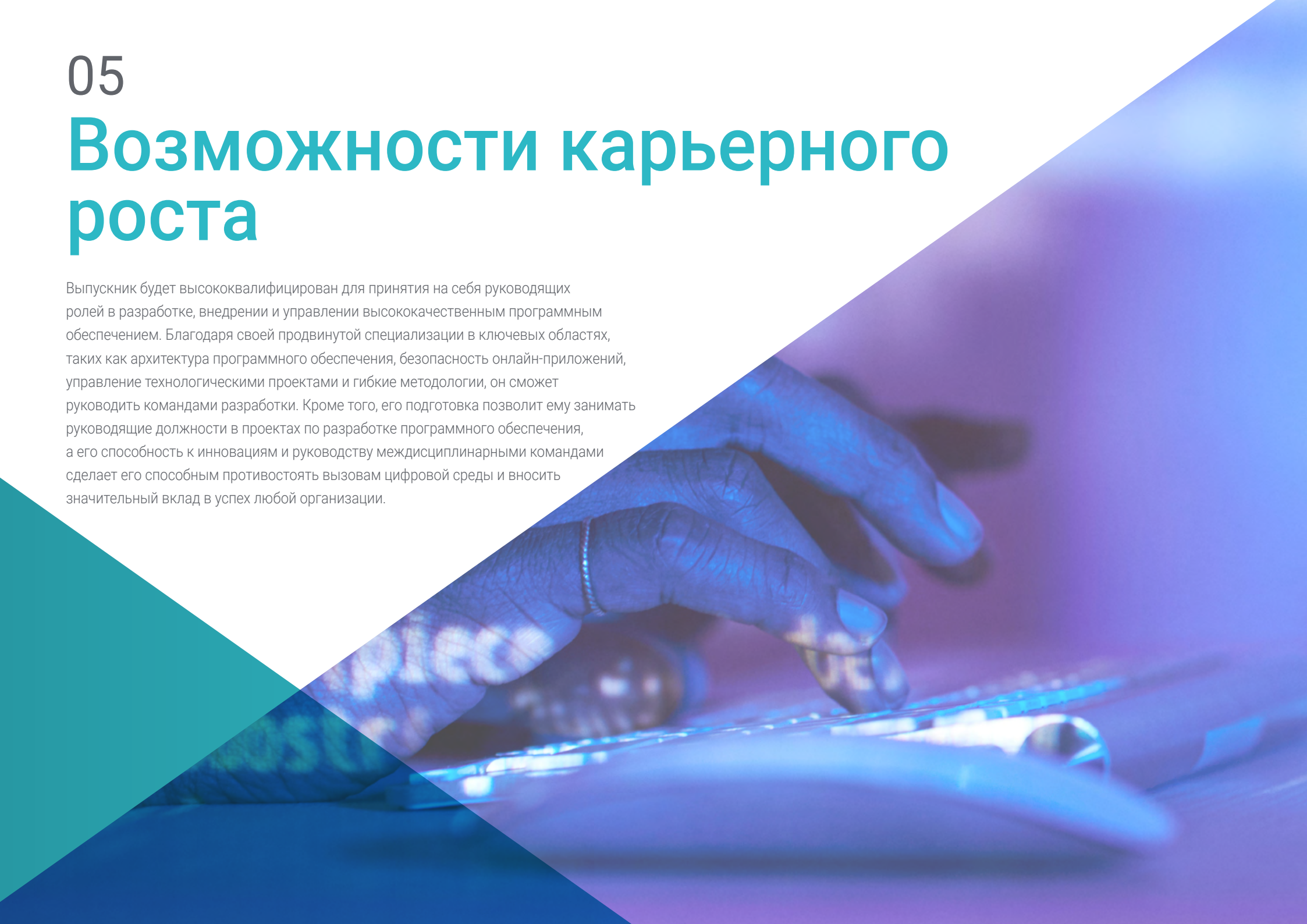
Модуль 20. Безопасность в онлайн-приложениях

- Реализовывать решения для защиты онлайн-приложений от внешних и внутренних угроз
- Устанавливать политики безопасности и аудита для обеспечения целостности онлайн-приложений

05

Возможности карьерного роста

Выпускник будет высококвалифицирован для принятия на себя руководящих ролей в разработке, внедрении и управлении высококачественным программным обеспечением. Благодаря своей продвинутой специализации в ключевых областях, таких как архитектура программного обеспечения, безопасность онлайн-приложений, управление технологическими проектами и гибкие методологии, он сможет руководить командами разработки. Кроме того, его подготовка позволит ему занимать руководящие должности в проектах по разработке программного обеспечения, а его способность к инновациям и руководству междисциплинарными командами сделает его способным противостоять вызовам цифровой среды и вносить значительный вклад в успех любой организации.



TECH дает вам возможность реализовать свои мечты в самой захватывающей дисциплине, которая превращает идеи в осязаемые продукты, способные улучшить жизнь людей”

Профиль выпускника

Профиль выпускника Профессиональной магистерской специализации по программной инженерии и качеству программного обеспечения ориентирован на подготовку высококвалифицированных специалистов, способных возглавлять и управлять технологическими проектами с высоким воздействием. Обеспечивая качество, безопасность и эффективность на всех этапах разработки программного обеспечения, он будет владеть как гибкими, так и традиционными методологиями. Кроме того, он будет готов к проектированию и разработке масштабируемых, эффективных и безопасных систем программного обеспечения, применяя международные стандарты качества и передовые методологии, такие как DevOps и непрерывная интеграция.

Станьте экспертом, который обеспечивает успех компаний, в крупнейшем цифровом университете мира.

- **Безопасность в программном обеспечении и системах:** Компетенция в реализации передовых мер безопасности, включая защиту данных и управление уязвимостями в онлайн-приложениях
- **Обеспечение качества программного обеспечения:** Умение применять международные стандарты (ISO, IEC 9126) и инструменты автоматизированного тестирования для гарантии надежности и производительности программного обеспечения
- **Разработка масштабируемых архитектур:** Способность проектировать и создавать системы программного обеспечения, которые могут расти и адаптироваться к требованиям рынка без ущерба для их качества и безопасности
- **Непрерывная интеграция и DevOps:** Умение внедрять и управлять процессами непрерывной интеграции, обеспечивая эффективную и бесперебойную поставку новых функциональностей программного обеспечения

После завершения Профессиональной магистерской специализации вы сможете применять свои знания и навыки на следующих должностях:

1. **Технический директор (CTO):** Ответственный за стратегическое управление технологиями в компании, возглавляющий команды разработки и контролирующий внедрение инновационных технологических решений.
2. **Менеджер по качеству программного обеспечения:** Ответственный за контроль и обеспечение соответствия процессов и продуктов программного обеспечения установленным стандартам качества, возглавляя инициативы по постоянному улучшению и тестированию программного обеспечения
3. **Архитектор программного обеспечения:** Главный разработчик структуры и архитектуры сложных систем программного обеспечения, обеспечивающий их масштабируемость, безопасность и эффективность
4. **Руководитель проектов по программному обеспечению:** Ответственный за планирование, выполнение и сдачу проектов программного обеспечения, управляя многопрофильными командами и обеспечивая, чтобы проекты были завершены в установленные сроки, в рамках бюджета и с соблюдением требуемых стандартов качества
5. **Специалист по информационной безопасности:** Ответственный за защиту приложений, инфраструктуры и данных от киберугроз, внедряющий эффективные стратегии и политики безопасности
6. **Аудитор безопасности программного обеспечения:** Проводит всесторонние аудиты для выявления уязвимостей в приложениях и системах, предлагая улучшения и решения для обеспечения безопасности программного обеспечения

“Если вы хотите изменить мир цифровых технологий к лучшему, выберите этот путь, который позволит вам стать экспертом в создании качественного программного обеспечения”

05

Методика обучения

TECH — первый в мире университет, объединивший метод ***кейс-стади*** с ***Relearning***, системой 100% онлайн-обучения, основанной на направленном повторении.

Эта инновационная педагогическая стратегия была разработана для того, чтобы предложить профессионалам возможность обновлять свои знания и развивать навыки интенсивным и эффективным способом. Модель обучения, которая ставит студента в центр учебного процесса и отводит ему ведущую роль, адаптируясь к его потребностям и оставляя в стороне более традиционные методологии.

TECH подготовит вас к решению новых задач в условиях неопределенности и достижению успеха в карьере”

Студент — приоритет всех программ TECH

В методике обучения TECH студент является абсолютным действующим лицом. Педагогические инструменты каждой программы были подобраны с учетом требований к времени, доступности и академической строгости, которые предъявляют современные студенты и наиболее конкурентоспособные рабочие места на рынке.

В асинхронной образовательной модели TECH студенты сами выбирают время, которое они выделяют на обучение, как они решат выстроить свой распорядок дня, и все это — с удобством на любом электронном устройстве, которое они предпочитают. Студентам не нужно посещать очные занятия, на которых они зачастую не могут присутствовать. Учебные занятия будут проходить в удобное для них время. Вы всегда можете решить, когда и где учиться.

“В TECH у вас НЕ будет занятий в реальном времени, на которых вы зачастую не можете присутствовать”

Самые обширные учебные планы на международном уровне

TECH характеризуется тем, что предлагает наиболее обширные академические планы в университетской среде. Эта комплексность достигается за счет создания учебных планов, которые охватывают не только основные знания, но и самые последние инновации в каждой области.

Благодаря постоянному обновлению эти программы позволяют студентам быть в курсе изменений на рынке и приобретать навыки, наиболее востребованные работодателями. Таким образом, те, кто проходит обучение в TECH, получают комплексную подготовку, которая дает им значительное конкурентное преимущество для продвижения по карьерной лестнице.

Более того, студенты могут учиться с любого устройства: компьютера, планшета или смартфона.

> *Модель TECH является асинхронной, поэтому вы можете изучать материал на своем компьютере, планшете или смартфоне в любом месте, в любое время и в удобном для вас темпе”*

Case studies или метод кейсов

Метод кейсов является наиболее распространенной системой обучения в лучших бизнес-школах мира. Разработанный в 1912 году для того, чтобы студенты юридических факультетов не просто изучали законы на основе теоретических материалов, он также имел цель представить им реальные сложные ситуации. Таким образом, они могли принимать взвешенные решения и выносить обоснованные суждения о том, как их разрешить. В 1924 году он был установлен в качестве стандартного метода обучения в Гарвардском университете.

При такой модели обучения студент сам формирует свою профессиональную компетенцию с помощью таких стратегий, как *обучение действием* (learning by doing) или *дизайн-мышление* (*design thinking*), используемых такими известными учебными заведениями, как Йель или Стэнфорд.

Этот метод, ориентированный на действия, будет применяться на протяжении всего академического курса, который студент проходит в TECH. Таким образом, они будут сталкиваться с множеством реальных ситуаций и должны будут интегрировать знания, проводить исследования, аргументировать и защищать свои идеи и решения. Все это делается для того, чтобы ответить на вопрос, как бы они поступили, столкнувшись с конкретными сложными событиями в своей повседневной работе.

Метод *Relearning*

В TECH *метод кейсов* дополняется лучшим методом онлайн-обучения — *Relearning*.

Этот метод отличается от традиционных методик обучения, ставя студента в центр обучения и предоставляя ему лучшее содержание в различных форматах. Таким образом, студент может пересматривать и повторять ключевые концепции каждого предмета и учиться применять их в реальной среде.

Кроме того, согласно многочисленным научным исследованиям, повторение является лучшим способом усвоения знаний. Поэтому в TECH каждое ключевое понятие повторяется от 8 до 16 раз в рамках одного занятия, представленного в разных форматах, чтобы гарантировать полное закрепление знаний в процессе обучения.

Метод Relearning позволит тебе учиться с меньшими усилиями и большей эффективностью, глубже вовлекаясь в свою специализацию, развивая критическое мышление, умение аргументировать и сопоставлять мнения — прямой путь к успеху.

Виртуальный кампус на 100% в онлайн-формате с лучшими учебными ресурсами

Для эффективного применения своей методики TECH предоставляет студентам учебные материалы в различных форматах: тексты, интерактивные видео, иллюстрации, карты знаний и др. Все они разработаны квалифицированными преподавателями, которые в своей работе уделяют особое внимание сочетанию реальных случаев с решением сложных ситуаций с помощью симуляции, изучению контекстов, применимых к каждой профессиональной сфере, и обучению на основе повторения, с помощью аудио, презентаций, анимации, изображений и т.д.

Последние научные данные в области нейронаук указывают на важность учета места и контекста, в котором происходит доступ к материалам, перед началом нового процесса обучения. Возможность индивидуальной настройки этих параметров помогает людям лучше запоминать и сохранять знания в гиппокампе для долгосрочного хранения. Речь идет о модели, называемой *нейрокогнитивным контекстно-зависимым электронным обучением*, которая сознательно применяется в данной университетской программе.

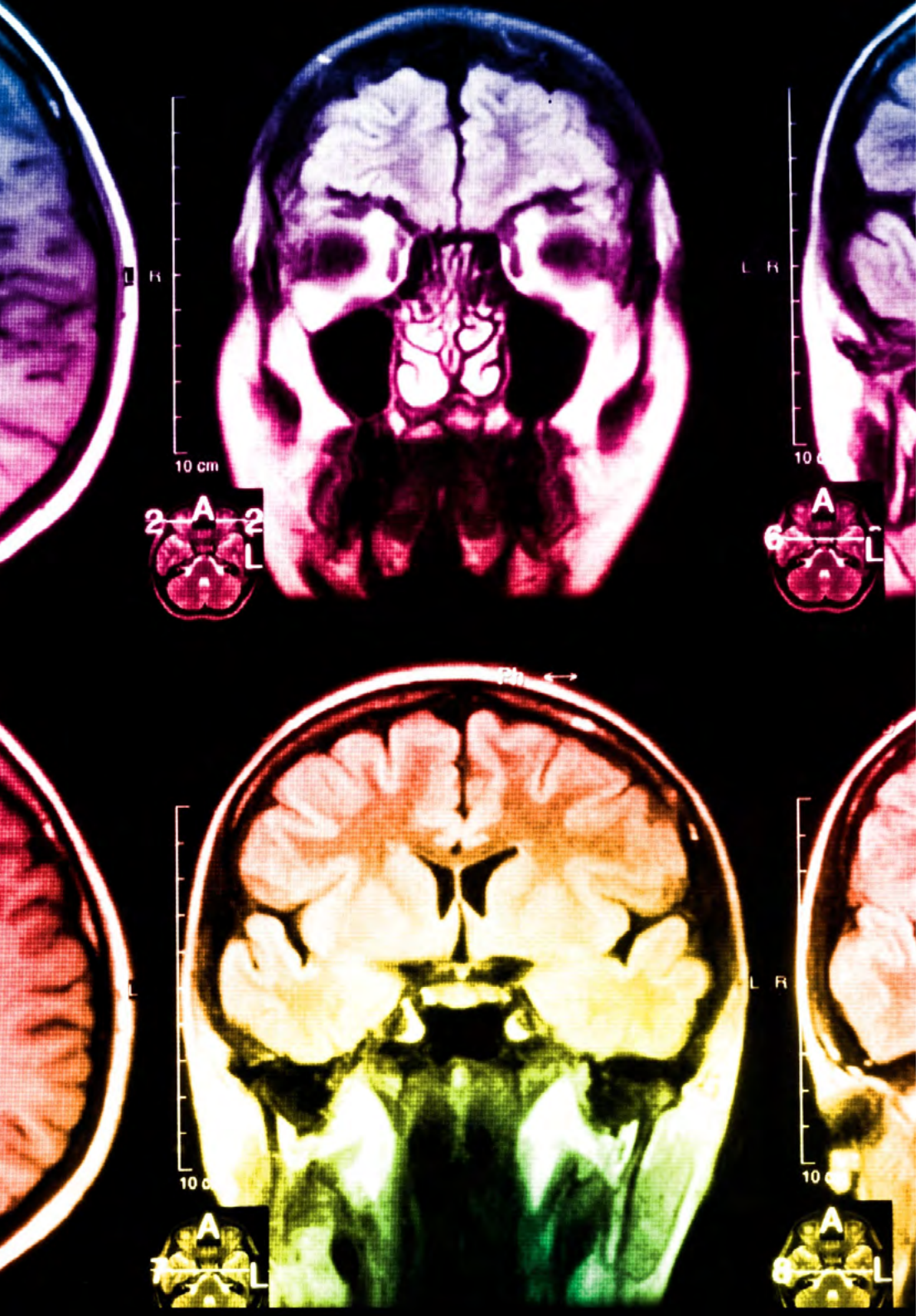
Кроме того, для максимального содействия взаимодействию между наставником и студентом предоставляется широкий спектр возможностей для общения как в реальном времени, так и в отложенном (внутренняя система обмена сообщениями, форумы для обсуждений, служба телефонной поддержки, электронная почта для связи с техническим отделом, чат и видеоконференции).

Этот полноценный Виртуальный кампус также позволит студентам TECH организовывать свое учебное расписание в соответствии с личной доступностью или рабочими обязательствами. Таким образом, студенты смогут полностью контролировать академические материалы и учебные инструменты, необходимые для быстрого профессионального развития.

> *Онлайн-режим обучения на этой программе позволит вам организовать свое время и темп обучения, адаптировав его к своему расписанию”*

Эффективность метода обосновывается четырьмя ключевыми достижениями:

1. Студенты, которые следуют этому методу, не только добиваются усвоения знаний, но и развивают свои умственные способности с помощью упражнений по оценке реальных ситуаций и применению своих знаний.
2. Обучение прочно опирается на практические навыки, что позволяет студенту лучше интегрироваться в реальный мир.
3. Усвоение идей и концепций становится проще и эффективнее благодаря использованию ситуаций, возникших в реальности.
4. Ощущение эффективности затраченных усилий становится очень важным стимулом для студентов, что приводит к повышению интереса к учебе и увеличению времени, посвященному на работу над курсом.



Методика университета, получившая самую высокую оценку среди своих студентов

Результаты этой инновационной академической модели подтверждаются высокими уровнями общей удовлетворенности выпускников TECH.

Студенты оценивают качество преподавания, качество материалов, структуру и цели курса на отлично. Неудивительно, что учебное заведение стало лучшим университетом по оценке студентов на платформе отзывов Global Score получив 4,9 балла из 5.

Благодаря тому, что TECH идет в ногу с передовыми технологиями и педагогикой, вы можете получить доступ к учебным материалам с любого устройства с подключением к Интернету (компьютера, планшета или смартфона).

Вы сможете учиться, пользуясь преимуществами доступа к симулированным образовательным средам и модели обучения через наблюдение, то есть учиться у эксперта (learning from an expert).

Таким образом, в этой программе будут доступны лучшие учебные материалы, подготовленные с большой тщательностью:

Учебные материалы

Все дидактические материалы создаются преподавателями специально для студентов этого курса, чтобы они были действительно четко сформулированными и полезными.

Затем эти материалы переносятся в аудиовизуальный формат, на основе которого строится наш способ работы в интернете, с использованием новейших технологий, позволяющих нам предложить вам отличное качество каждого из источников, предоставленных к вашим услугам.

Практика навыков и компетенций

Студенты будут осуществлять деятельность по развитию конкретных компетенций и навыков в каждой предметной области. Практика и динамика приобретения и развития навыков и способностей, необходимых специалисту в рамках глобализации, в которой мы живем.

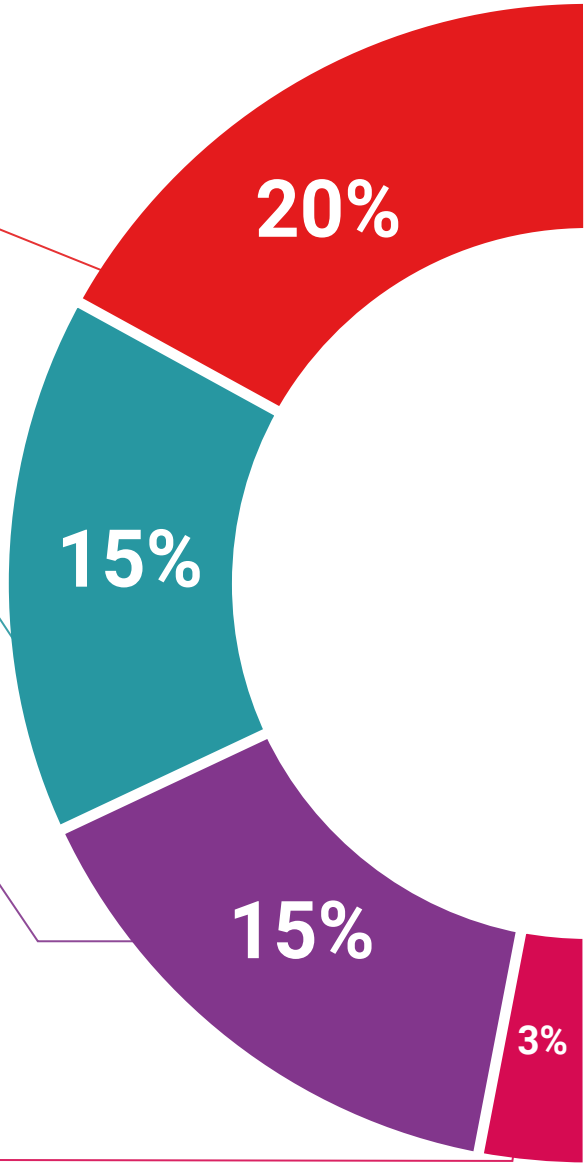
Интерактивные конспекты

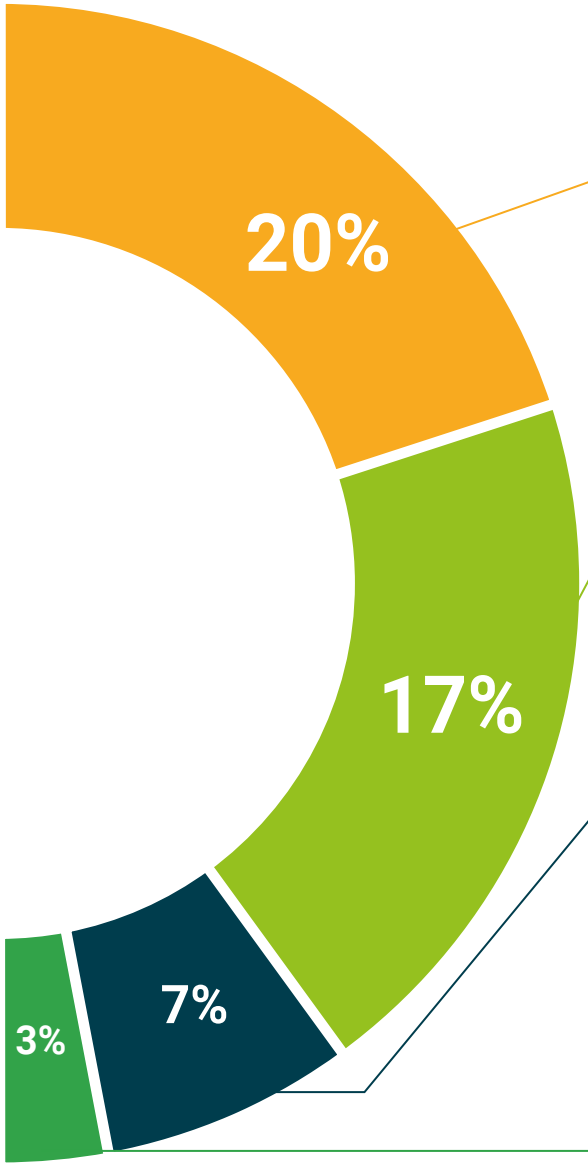
Мы представляем содержание в привлекательной и динамичной форме для воспроизведения на мультимедийных устройствах, которые включают аудио, видео, изображения, диаграммы и концептуальные карты для закрепления знаний.

Эта эксклюзивная образовательная система для презентации мультимедийного содержания была награждена Microsoft как "Кейс успеха в Европе".

Дополнительная литература

Последние статьи, консенсусные документы, международные рекомендации... В нашей виртуальной библиотеке вы получите доступ ко всему, что необходимо для прохождения обучения.





Кейс-стади

Студенты завершат выборку лучших *кейс-стади* по предмету. Кейсы представлены, проанализированы и преподаются ведущими специалистами на международной арене.

Тестирование и повторное тестирование

Мы периодически оцениваем и переоцениваем ваши знания на протяжении всей программы. Мы делаем это на 3 из 4 уровней пирамиды Миллера.

Мастер-классы

Существуют научные данные о пользе экспертного наблюдения третьей стороны.

Так называемый метод *обучения у эксперта* (learning from an expert) укрепляет знания и память, а также формирует уверенность в ваших будущих сложных решениях.

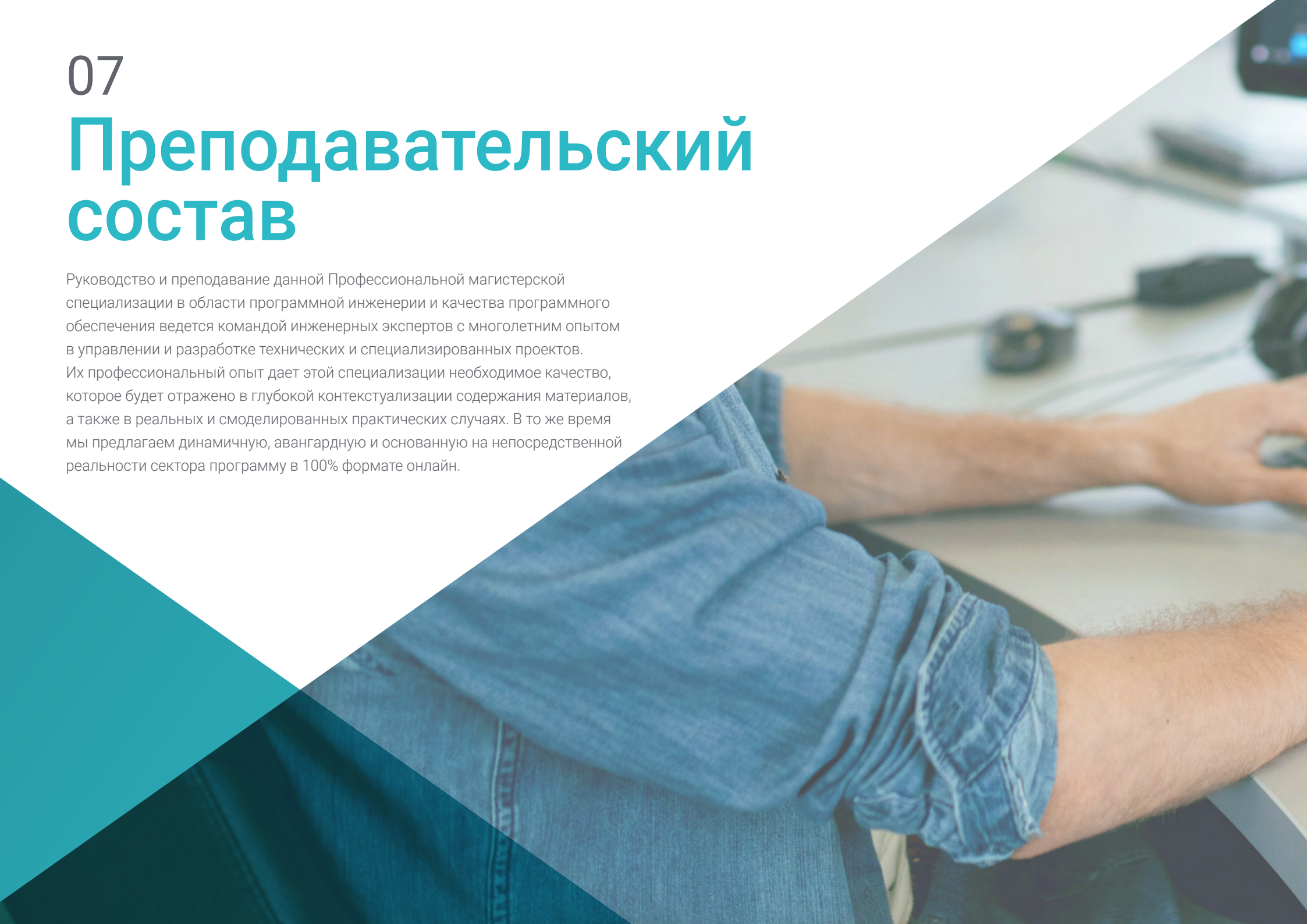
Краткие справочные руководства

TECH предлагает наиболее актуальные материалы курса в виде карточек или кратких справочных руководств. Это сжатый, практичный и эффективный способ помочь студенту продвигаться в обучении.

07

Преподавательский состав

Руководство и преподавание данной Профессиональной магистерской специализации в области программной инженерии и качества программного обеспечения ведется командой инженерных экспертов с многолетним опытом в управлении и разработке технических и специализированных проектов. Их профессиональный опыт дает этой специализации необходимое качество, которое будет отражено в глубокой контекстуализации содержания материалов, а также в реальных и смоделированных практических случаях. В то же время мы предлагаем динамичную, авангардную и основанную на непосредственной реальности сектора программу в 100% формате онлайн.



Присоединяйтесь к самой востребованной карьере для специалистов по разработке и качеству программного обеспечения и воспользуйтесь возможностью стать одним из них"

Приглашенный лектор международного уровня

Даррен Палсифер - опытный **архитектор программного обеспечения**, новатор с выдающимся международным послужным списком в области **разработки программного обеспечения и** микропрограмм. Кроме того, он обладает высокоразвитыми навыками **общения**, **управления проектами** и ведения **бизнеса**, что позволило ему возглавить крупные глобальные инициативы.

На протяжении своей карьеры он также занимал ответственные должности, такие как **главный архитектор решений для государственного** сектора в корпорации **Intel**, где он продвигал современный бизнес-**процессы и технологии** для клиентов, партнеров и пользователей в государственном секторе. Даррен основал компанию **Yoly Inc.**, где также занимал пост **генерального директора**, занимаясь созданием **инструментов для агрегации и диагностики социальных сетей** на основе **программного обеспечения как услуга (SaaS)**, использующее технологии ***больших данных*** и **Веб 2**.0.

Даррен работал в других компаниях, в том числе **генеральным директором по инженерным** вопросам в **Dell Technologies**, где возглавлял подразделение ***больших данных* в облаке**, руководил командами в **США** и **Китае** по управлению крупными проектами и реструктуризации бизнес-подразделений для успешной интеграции. Он также занимал должность директора по информационным технологиям (***Chief Information Officer*** в компании XanGo, где руководил такими проектами, как поддержка справочной службы, **поддержка производства и разработка решений**.

Среди множества специализаций, в которых он является экспертом, выделяется технология ***Edge to Cloud***, **кибербезопасность**, **генеративный искусственный интеллект**, ***разработка программного обеспечения***, **сетевые технологии**, **облачная нативная разработка** и **контейнерная экосистема**. Он делится своими знаниями в еженедельном **подкасте** и **информационном бюллетене** "***Embracing Digital Transformation***", который он создал и представил, помогая организациям успешно пройти через **цифровую трансформацию**, используя **персонал**, **процессы** и **технологии**.

Г-н Палсифер, Даррен

- Главный архитектор решений для государственного сектора, Intel, Калифорния, США
- Ведущий и автор проекта *"Воплощение цифровой трансформации"*, Калифорния, США
- Основатель и генеральный директор компании Yoly Inc., Арканзас
- Генеральный директор по инженерным вопросам в компании Dell Technologies, Арканзас
- Директор по информационным технологиям (*Chief Information Officer*) компании XanGo, Юта
- Старший архитектор в Cadence Design Systems, Калифорния
- Старший менеджер по проектным процессам в Lucent Technologies, Калифорния
- Инженер-программист в компании Cemax-Icon, Калифорния
- Инженер-программист в компании ISG Technologies, Канада
- MBA в области управления технологиями в Университете Феникса, Калифорния
- Степень бакалавра в области информатики и электротехники в Университете Бригама Янга

Благодаря TECH вы сможете учиться у лучших мировых профессионалов"

Руководство

Г-н Молина Молина, Херонимо

- Руководитель отдела искусственного интеллекта в Helphone
- Инженер искусственного интеллекта и архитектор программного обеспечения в NASSAT, спутниковый интернет в движении
- Старший консультант в компании Hexa Ingenieros
- Внедрение искусственного интеллекта (ML и CV)
- Эксперт по решениям на основе искусственного интеллекта в области компьютерного зрения, ML/DL и NLP
- Курс профессиональной подготовки в области создания и развития бизнеса в Bancaixa и Fundeun
- Компьютерный инженер из Университета Аликанте
- Степень магистра в области искусственного интеллекта в Католическом университете Авилы
- Руководитель MBA на Европейском форуме бизнес-кампусов

Преподаватели

Г-жа Родригес Мигес, Кандида

- Младший разработчик приложений в Getronics
- Сооснователь и руководитель городской сети искусственного интеллекта в Галисии
- Младший инженер-программист в Indra
- Веб-разработчик в EDISA
- Степень бакалавра компьютерной инженерии Университета Виго
- Степень магистра в области компьютерной инженерии в Университете Виго

Г-н Пи Морель, Ориоль

- Функциональный аналитик в компании Fihoca
- Владелец продуктов хостинга и электронной почты в CDmon
- Функциональный аналитик и инженер-программист в компаниях Atmira и Capgemini
- Преподаватель в Capgemini, Forms Capgemini и Atmira
- Степень бакалавра в области технической инженерии по компьютерному управлению Автономного университета Барселоны
- Степень магистра в области искусственного интеллекта в Католическом университете Авилы
- MBA в области управления и администрирования бизнеса от IMF Smart Education
- Степень магистра в области управления информационными системами от IMF Smart Education
- Аспирантура в области шаблонов проектирования Открытого Университета Каталонии

Г-н Мартинес Кальво, Франсиско Хавьер

- Промышленный инженер специализация в области электроэнергетики и электроники
- Специалист по программному обеспечению в HEXA Ingenieros
- Старший разработчик .Net / архитектор решений Net в Everis
- Аналитик/архитектор *программного обеспечения* в LaLiga
- Выездной инженер Microsoft в BBVA
- *Внештатный* консультант по техническим вопросам и информатике
- Преподаватель Visual Studio, SqlServer, CCNA (маршрутизаторы и коммутаторы Cisco), PHP и .Net Web-программирование в нескольких центрах (Salesianos, Maforem, Dreamsoft)
- Промышленный инженер со специализацией в области электричества, промышленной электроники
- Степень магистра Cibernos в .NET, MCAD
- Степень магистра в области продвинутого программирования, экспертный уровень
- Степень магистра в области Web с сертификатами Dreamweaver, Fireworks, Flash и ActionScript, версии MX

Г-н Тенреро Моран, Маркос

- Инженер DevOps в Allot Communications
- Менеджер по управлению жизненным циклом приложений в Cegid Meta4
- Инженер по автоматизации QA в Cegid Meta4
- Степень магистра в области профессиональной разработки приложений для Android в Университете Галилео. Гватемала
- Степень магистра в области разработки облачных сервисов, Node.Js, JavaScript, HTML5 в Мадридском политехническом университете
- Веб-разработка с использованием Angular-CLI (4), Ionic и Node.Js, Meta4 Университета короля Хуана Карлоса
- Степень бакалавра компьютерной инженерии Университета короля Хуана Карлоса

Г-жа Асебес Тамарго, Патрисия

- Консультант, специализирующийся на больших данных
- Операционный отдел, работает с Elasticsearch и Kivana в компании Sirt
- Онлайн-исследователь в области *человеческого фактора* и *применения* искусственного интеллекта в Технологическом центре CTIC
- Онлайн-исследователь, Бизнес-подразделение в Технологическом центре CTIC
- Отдел цифрового здоровья и активного старения в Технологическом центре CTIC
- Отдел науки о данных в Технологическом центре CTIC
- Степень доктора в области компьютерных наук по искусственному интеллекту Политехнического университета Валенсии
- Степень бакалавра экономики в Университете Овьедо
- Степень магистра в области анализа данных UCJC
- Степень магистра в области исследований искусственного интеллекта в UNED
- Степень магистра в области *блокчейна, смарт-контрактов* и криптовалют Университета Алькала
- Последипломное образование в области *блокчейн-инженерии* в EADA
- Степень магистра в области экономики, инструментов, экономического анализа Университета Овьедо
- Степень магистра в области налогообложения в Colegio de Economistas (Ассоциация экономистов)

08

Квалификация

Профессиональная магистерская специализация в области программной инженерии и качества программного обеспечения гарантирует, помимо самого строгого и современного обучения, получение диплома о прохождении Профессиональной магистерской специализации, выдаваемого TECH Технологическим университетом.

Успешно пройдите эту программу и получите университетский диплом без хлопот, связанных с поездками и бумажной волокитой”

Данная **Профессиональной магистерской специализации в области программной инженерии и качества программного обеспечения** содержит самую полную и современную программу на рынке.

После прохождения аттестации студент получит по почте* с подтверждением получения соответствующий диплом **Профессиональной магистерской специализации,** выданный **TECH Технологическим университетом.**

Диплом, выданный **TECH Технологическим университетом**, подтверждает квалификацию, полученную в Профессиональной магистерской специализации, и соответствует требованиям, обычно предъявляемым биржами труда, конкурсными экзаменами и комитетами по оценке карьеры.

Диплом: **Профессиональной магистерской специализации в области программной инженерии и качества программного обеспечения**

Формат: **онлайн**

Продолжительность: **2 года**


tech технологический университет

Настоящий

ДИПЛОМ

свидетельствует о том, что

Г-н/Г-жа _____, с документом, удостоверяющий личность_____

Освоил(а) и успешно прошел(ла) аккредитацию программы

ПРОФЕССИОНАЛЬНАЯ МАГИСТЕРСКАЯ СПЕЦИАЛИЗАЦИЯ

в области

программной инженерии и качества программного обеспечения

Данный диплом специализированной программы, присуждаемый Университетом, соответствует 3000 учебным часам, с датой начала дд/мм/гггг и датой окончания дд/мм/гггг.

TECH является частным высшим учебным заведением, признанным Министерством народного образования Мексики с 28 июня 2018 года.

17 июня 2020 г.

Д-р Tere Guevara Navarro
Ректор

unique TECH code: AFWOR23S techtitute.com/certificates


Профессиональная магистерская специализация в области программной инженерии и качества программного обеспечения

Общее распределение учебного плана

Курс	Предмет	Часы	Характер
1º	Качество программного обеспечения. Уровни развития TRL	150	Обязат
1º	Разработка программных проектов. Функциональная и техническая документация.	150	Обязат
1º	Тестирование программного обеспечения. Автоматизация тестирования	150	Обязат
1º	Методологии управления проектами программного обеспечения. Каскадная модель vs agile-методологии	150	Обязат
1º	TDD (Test Driven Development). Разработка программного обеспечения через тестирование	150	Обязат
1º	DevOps. Управление качеством программного обеспечения	150	Обязат
1º	DevOps и непрерывная интеграция. Передовые практические решения в разработке программного обеспечения	150	Обязат
1º	Проектирование баз данных (БД). Стандартизация и производительность. Качество программного обеспечения	150	Обязат
1º	Проектирование масштабируемых архитектур. Архитектура в жизненном цикле программного обеспечения	150	Обязат
2º	Критерии качества ISO, IEC 9126. Метрики качества программного обеспечения	150	Обязат
2º	Методологии, разработка и качество в программной инженерии	150	Обязат
2º	Управление программного обеспечения проекта	150	Обязат
2º	Платформы для разработки программного обеспечения	150	Обязат
2º	Вычисления на стороне клиента в веб-разработке	150	Обязат
2º	Вычисления на стороне веб-сервера	150	Обязат
2º	Управление безопасностью	150	Обязат
2º	Безопасность программного обеспечения	150	Обязат
2º	Администрирование веб-сервера	150	Обязат
2º	Аудит безопасности	150	Обязат
2º	Безопасность в онлайн-приложениях	150	Обязат

Д-р Tere Guevara Navarro
Ректор

tech технологический университет

*Гаагский апостиль. В случае, если студент потребует, чтобы на его диплом в бумажном формате был проставлен Гаагский апостиль, TECH EDUCATION предпримет необходимые шаги для его получения за дополнительную плату.

tech технологический университет

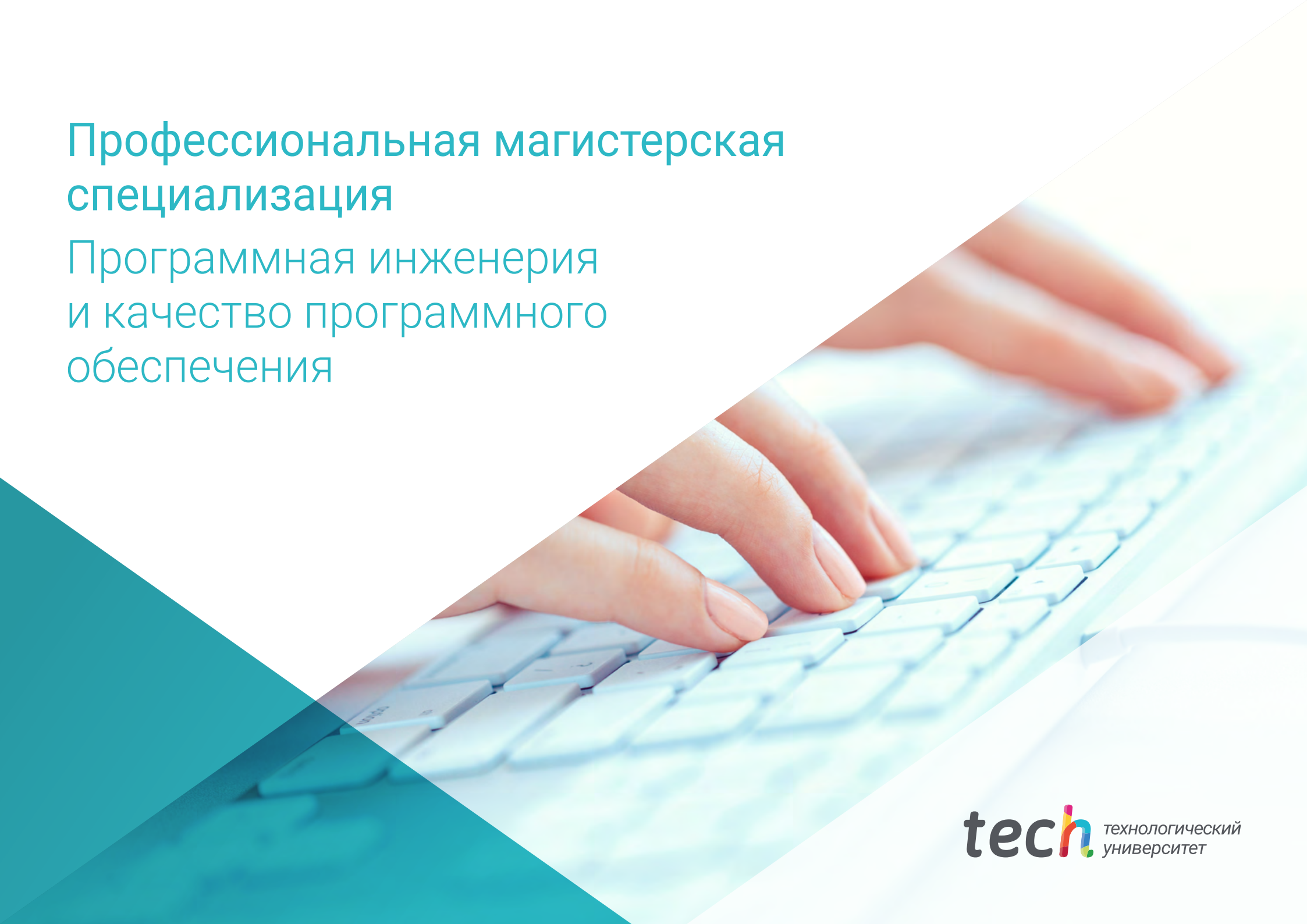
Профессиональная магистерская специализация

Программная инженерия и качество программного обеспечения

- Формат: **онлайн**
- Продолжительность: **2 года**
- Учебное заведение: **TECH Технологический университет**
- Расписание: **по своему усмотрению**
- Экзамены: **онлайн**

Профессиональная магистерская специализация

Программная инженерия и качество программного обеспечения



tech технологический университет