

Буценко Максим Сергеевич

студент

Дозоренко Михаил Сергеевич

студент

Трандасир Артем Сергеевич

студент

Никитин Давид Дмитриевич

студент

Научный руководитель

Куликова Наталья Николаевна

канд. биол. наук, доцент

ФГБОУ ВО «Кубанский государственный университет»

г. Краснодар, Краснодарский край

ИНТЕГРАЦИЯ ПРАКТИК DEVOPS, GAMEDEV И COMPUTER VISION В ПРОФЕССИОНАЛЬНУЮ ПОДГОТОВКУ ИНЖЕНЕРОВ-ПРОГРАММИСТОВ

***Аннотация:** в статье рассматривается интеграция трёх прикладных направлений – DevOps, GameDev и компьютерного зрения (Computer Vision) – в профессиональную подготовку инженеров-программистов. Названные направления трактуются как взаимодополняющие инженерные практики, освоение которых целесообразно вести сквозным образом, в рамках единого прикладного проекта. Предложена модель такого проекта, объединяющего автоматизацию реагирования на оповещения, разработку в среде Unity Editor и применение моделей компьютерного зрения. Подход апробирован при подготовке магистрантов: перенос знаний из различных дисциплин в общий прикладной контекст заметно повышает инженерную готовность выпускников.*

***Ключевые слова:** профессиональное образование, цифровая трансформация, подготовка программистов, DevOps, автоматическое реагирование,*

GameDev, Unity Editor, компьютерное зрение, маска захвата, сквозной учебный проект.

Введение.

Современный рынок труда в сфере информационных технологий предъявляет к инженеру-программисту требования, выходящие за рамки отдельной академической дисциплины. Выпускник должен одновременно владеть базовыми принципами автоматизации эксплуатации (DevOps), уметь разрабатывать приложения в специализированных интегрированных средах, а также понимать принципы машинного обучения и, в частности, компьютерного зрения [1; 2]. При этом образовательные программы по-прежнему выстраиваются вокруг изолированных дисциплин, а межпредметные связи проявляются, как правило, лишь на уровне выпускной квалификационной работы, что негативно сказывается на практической готовности выпускника.

Цель статьи – обосновать целесообразность интеграции трёх прикладных направлений (DevOps, GameDev, Computer Vision) в подготовку инженеров-программистов и предложить модель сквозного учебного проекта, в котором эти направления образуют единый прикладной контекст. Под сквозным проектом авторы понимают практическую работу, в рамках которой результаты освоения трёх различных дисциплин объединяются в едином программном продукте.

DevOps как практика автоматизации реагирования.

DevOps объединяет разработку и эксплуатацию программных систем. С точки зрения подготовки инженера-программиста среди соответствующих практик особенно значимы непрерывная интеграция, мониторинг и автоматическое реагирование на инциденты [3]. Учебный материал, в котором обучающийся получает не только теоретические сведения о CI/CD, но и опыт построения подсистемы, автоматически обрабатывающей входящие оповещения, формирует у него целостное представление о жизненном цикле продукта в эксплуатационной среде.

В качестве методического приёма мы рассматриваем разработку обучающимся модуля автоматического реагирования: подсистема получает оповещения

от средств мониторинга, классифицирует их и активирует заранее описанные сценарии устранения инцидентов. Базовая конфигурация маршрутизации оповещений на автоматизированный обработчик приведена на рис. 1.

```
1 # alertmanager: автоматическое устранение инцидента
2 route:
3   group_by: ['alertname', 'service']
4   receiver: 'auto-remediation'
5
6 receivers:
7   - name: 'auto-remediation'
8     webhook_configs:
9       - url: 'http://remediator.svc/v1/handle'
10       send_resolved: true
11
12 # remediator (handler):
13 # 1) сопоставляет alert.labels с runbook
14 # 2) выполняет команды восстановления (restart, scale, rotate)
15 # 3) фиксирует исход в journal, при провале эскалирует дежурному
```

Рис. 1. Минимальная конфигурация маршрутизации оповещений на автоматизированный обработчик

GameDev и среда Unity Editor.

GameDev нередко воспринимается как обособленная индустрия, однако в контексте профессиональной подготовки он предоставляет методически удобную проектную среду. Unity Editor сочетает редактор сцены, систему ассетов, среду исполнения и инструментарий расширения через плагины, что позволяет обучающемуся проследить полный путь от ресурса проекта до запускаемого приложения [4].

Для подготовки инженера-программиста в среде Unity Editor значимы две стороны: использование редактора как платформы прикладной разработки и расширение редактора собственными инструментами. Второе направление позволяет обучающемуся работать с архитектурными темами, наблюдателями, реактивными потоками, автоматическими обработчиками ресурсных событий), привычными в общей инженерной практике, но в наглядной среде с понятным визуальным результатом.

Computer Vision как прикладной модуль.

Компьютерное зрение – одно из наиболее востребованных направлений прикладного машинного обучения. В подготовке инженера-программиста ключ-

чевым является не построение новых архитектур, а применение готовых сегментационных и детекционных моделей: подготовка входных данных, выбор бэкенда инференса, формирование маски целевого объекта и генерация прикладных событий по результатам сегментации [5].

В учебной программе CV-модуль методически удобен тем, что выходные данные модели – маска или ограничивающий прямоугольник – служат входами для других подсистем. Тем самым компьютерное зрение естественным образом связывается как с GameDev (визуализация маски в Unity-сцене), так и с DevOps (генерация программных событий, по которым автоматически активируются заранее описанные реакции).

Сквозной учебный проект как интегрирующий элемент.

Объединение трёх направлений достигается за счёт формулировки сквозного учебного проекта – учебного тренажёра, в котором CV-маска формирует событие, активирующее автоматическое реагирование в Unity-сцене. Схема такой интеграции показана на рис. 2.

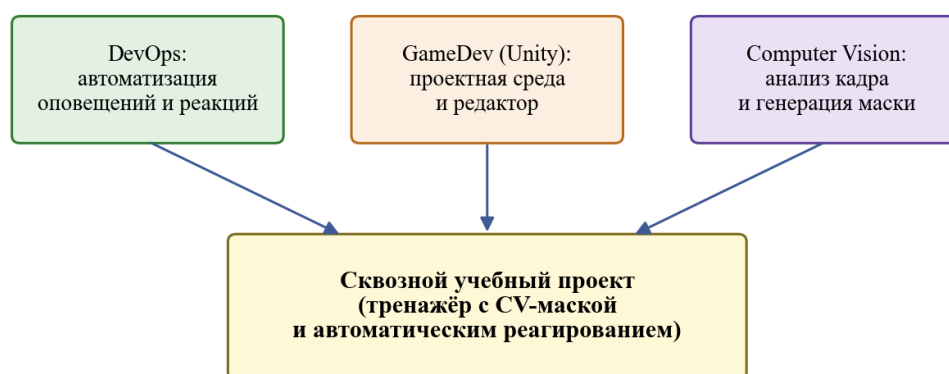


Рис. 2. Интеграция прикладных практик в подготовку инженера-программиста

Совмещение трёх направлений в одном программном продукте даёт обучающемуся опыт инженерной интеграции. В лекционных курсах этот навык практически не отрабатывается, и в стандартной образовательной траектории выпускник приобретает его уже на рабочем месте. Дополнительный эффект состоит

в том, что освоенные технологии оказываются связаны общим прикладным контекстом и переносятся между задачами без дополнительных методических усилий.

Заключение.

Профессиональная подготовка инженера-программиста требует переноса знаний между дисциплинами в общий прикладной проект. Интеграция практик DevOps, GameDev и компьютерного зрения через сквозной учебный проект даёт магистранту опыт сразу в трёх направлениях – событийная автоматизация, прикладная разработка в специализированной среде и инференс моделей машинного обучения. Предложенная модель воспроизводима и применима в подготовке магистрантов профильных направлений.

Список литературы

1. Минин М.Г. Проектирование образовательной программы подготовки инженеров в условиях цифровой трансформации / М.Г. Минин, Н.С. Михайлова // Высшее образование в России. – 2021. – Т. 30, №8/9. – С. 117–127.
2. Профессиональный стандарт «Программист»: утв. приказом Минтруда России от 20.07.2022 №424н. Зарегистрирован в Минюсте России 17.08.2022 №69633.
3. Kim G. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations / G. Kim, J. Humble, P. Debois, J. Willis. 2nd ed. Portland: IT Revolution Press, 2021. 480 p.
4. Unity User Manual / Unity Technologies. URL: <https://docs.unity3d.com/Manual/index.html> (date of request: 12.05.2026).
5. Шапиро Л. Компьютерное зрение / Л. Шапиро, Дж. Стокман; пер. с англ. – 4-е изд. – М.: БИНОМ. Лаборатория знаний, 2020. – 752 с.