

Nvidia GeForce RTX 3080 12GB.

В качестве метрик оценки производительности использовались длительность обучения и точность. Результаты сравнения представлены в таблице 1.

Таблица 1 – Сравнение точности и скорости обучения

Метрика	Число блоков					
	2	4	6	8	10	12
Точность, %	64,94	67,53	67,91	69,21	68,1	67,51
Длительность обучения, сек.	149,36	194,38	249,77	296,03	333,17	364,29

Как показывают результаты, максимальная точность достигается при использовании 8 блоков MLP-Mixer. Скорость обучения при этом остаётся на приемлемом уровне.

Таким образом, оптимальное число блоков для данной задачи составляет 8, что обеспечивает баланс между точностью и временем обучения.

Список использованных источников

1. MLP-Mixer: An all-MLP Architecture for Vision [Electronic resource] / Ilya T., Neil H., Alexander K., Lucas B., Xiaohua Z., Thomas U., Jessica Y., Andreas S., Daniel K., Jakob U., Mario L., Alexey D. // Cornell University. – 2021. – Mode of access: <https://arxiv.org/pdf/2105.01601/>. – Date of access: 07.03.2024.

УДК 004.05

МЕТОДЫ ОЦЕНКИ ЭФФЕКТИВНОСТИ ПРОЦЕССА РАЗРАБОТКИ

Знатнов Г. Д., студ., Каргина Л. Р., д.э.н., проф.

Российский университет транспорта, г. Москва, Российская Федерация

Реферат. В статье рассмотрены теоретические основы оценки эффективности разработки, разобраны две методологии DORA и SPACE. Оценка эффективности разработки имеет важное значение, так как от этого напрямую зависят успехи любой IT-компаний, их финансовые показатели и бизнес-результаты.

Ключевые слова: эффективность разработки, производительность, разработка, эффективность, фреймворки, SPACE, DORA, метрики эффективности, IT-технологии.

Сфера IT-технологий является одной из самой крупных в современном бизнесе – по прогнозам компании Forrester общемировые расходы на IT-технологии в 2025 году поднимутся до значения 5,6 % и достигнут \$ 4,9 трлн. Важнейшими факторами роста являются активное внедрение программного обеспечения и IT-услуг во все сферы жизни, высокий спрос и развитие искусственного интеллекта, а также облачных технологий. IT-компании вкладывают и зарабатывают огромные деньги, именно поэтому поднимается спрос на любые методы оптимизации производительности бизнес-процессов – для получения наибольшей выручки. Так как разработчики являются важнейшим механизмом создания IT-продукта, инициативы по улучшению эффективности процесса разработки выходят на первый план.

Эффективность разработки заключается не в том, чтобы производить как можно больше программного кода, а в том насколько эффективно этот код был написан и насколько качественно и быстро он приводит к достижению бизнес-результатов компании. Однако в этом и заключается проблема: процесс разработки является очень комплексным и не может быть оценен количественно, в отличие от маркетинга или сферы продаж, где все можно измерить цифрами. Таким образом, в измерении эффективности разработки важно оценивать как количественные, так и качественные показатели.

В этой статье я рассматриваются существующие метрики оценки эффективности труда разработчиков, факторы, которые влияют на разработку, возможные подводные камни при оценке производительности труда.

Эффективность разработки или производительность разработчика – это показатель, который отражает насколько рационально был создан IT-продукт с учетом таких аспектов как достигнутый результат, качество и затраты:

Эффективность разработки затрагивает следующие факторы.

1. Темп разработки: скорость, с которой разработчики пишут код и закрывают задачи.
2. Качество кода: способность писать наиболее релевантный, точный и не содержащий багов код.
3. Коммуникация: способность взаимодействовать с членами команды, обсуждать проблемы, запрашивать помощь в случае возникновения трудностей.
4. Трудовая отдача: умение содержать себя в наиболее трудоспособной форме, способность наиболее эффективно использовать свое время для выполнения задач и заинтересованность в получении качественного результата. Стремление бросать вызовы и справляться с ними.
5. Следование лучшим практикам: умение использовать передовые инструменты, а также способность и стремление использовать наиболее актуальные технологии в разработке.

Кроме того, на эффективность разработки могут влиять и сторонние факторы, такие как бизнес-процессы компании, инструменты, которые нормативно обязательны к использованию, уровень команды, с которой взаимодействует разработчик, условия труда, атмосфера в команде и другие.

Рассмотрим существующие метрики оценки эффективности:

DORA (DevOps Research and Assessment) – фреймворк DORA состоит из четырех показателей, которые разделяют на две группы Throughput (Пропускная способность) и Stability (Стабильность).

1. Throughput измеряет частоту происходящих изменений. Включает следующие метрики:
 - Deployment Frequency или частота развертывания – данная метрика показывает, как часто команда разработки выпускает изменения в прод-сборку. Чем чаще, тем лучше.
 - Lead Time For Changes или время, необходимое для внесения изменений. Отвечает за время от коммита до его внесения в прод. Чем чаще, тем лучше.
2. Stability оценивает качество внесенных изменений и способность команды устранять сбои. Состоит из следующих метрик:
 - Change Failure Rate – частота развертываний, которые приводят к возникновению ошибок и проблем в проде. Чем меньше, тем лучше.
 - Time To Restore Services или время, необходимое для восстановления обслуживания. Отвечает за время, которое необходимо для восстановления обслуживания после ошибки, которая привела к поломкам на проде. Чем меньше, тем лучше.

Регулярно измеряя данные показатели, можно получить оценку уровня DevOps и следить за прогрессом команды. Оценка по каждому показателю классифицируется на Elite, High, Medium, Low.

Другой фреймворк – **SPACE** был создан GitHub совместно с Microsoft. Данная концепция может быть полезным дополнением для DORA, так как в отличие от DORA, данный фреймворк больше сфокусирован на качественных показателях, а не количественных. Аспекты, которые рассматривает данная метрика спрятаны в названии.

1. Satisfaction and Well-being – удовлетворение и благополучие разработчиков, которое они получают от работы. Данный показатель напрямую влияет на производительность всей команды. Оценить удовлетворение от работы можно, проводя опросы.
2. Performance или производительность оценивается на основе результатов разработчика или команды. Оценивать данный критерий сложно, делать это нужно совместно с более опытным разработчиком, например, с Team Lead, так как здесь важно оценить не только количество написанного кода, но и его эффективность. Также можно оценивать время написания кода, количество багов, время их фиксов и другие.
3. Activity или активность включает в себя количество работы, которые разработчик выполнил в процессе разработки ПО. Данное значение показывает темп работы и количество полезных действий. Оценить данный показатель можно по полезным действиям, количеству Story Points, выполненным задачам и количеству документации.
4. Communication and Collaboration. Коммуникация и взаимодействие является очень важным аспектом и самым сложным для оценивания. Команда, у которой хорошо налажено общение внутри, однозначно будет более продуктивной, так как ее члены лучше знают, чем занимаются

их партнеры, осведомлены о статусах задач и приоритетах. В процессе оценивания данного аспекта можно смотреть на такие знаки, как скорость реагирования на задачи, паттерны взаимодействия членов команды, разрозненные области внутри команды, время, необходимое для адаптации новых сотрудников, осведомленность о задачах, качество документации.

5. **Efficiency and Flow.** Эффективность и Поток – данный показатель отображает насколько непрерывно ведется разработка. Другими словами, данный критерий показывает, насколько эффективно команда расходует время на разработку.

Таким образом, можно сказать, что обе методологии важны и позволяют оценить эффективность разработки. DORA больше сконцентрирована на количественных показателях, ее метрики понятны и выражены в цифрах – их можно сравнить. Фреймворк SPACE в свою очередь больше сосредоточен на качественных аспектах, затрагивает больше факторов, влияющих на разработку, в том числе социальных. Именно поэтому данный фреймворк является более сложным в применении.

При оценке производительности разработки могут возникать определенные проблемы. Например, некоторые метрики не могут быть полностью объективны без комплексного подхода. Так, например, неправильное восприятие критерия, отвечающего за часы, проведенные за компьютером, могут поспособствовать тому, что разработчики будут отдавать приоритет часам, а не написанию качественного кода. Также, разработчики могут писать избыточный код, чтобы повысить количество строк для создания иллюзии работы. Все это может привести к ситуации, когда вместо эффективности главной целью становится видимость занятости, а это в свою очередь может привести к неудовлетворительным результатам всей команды. Вышеперечисленные проблемы могут возникать, если подходить к оценке эффективности однобоко и не учитывать все возможные аспекты.

Кроме того, существует большое количество сторонних воздействий, которые также необходимо учитывать при оценке эффективности команды разработки. Сюда можно отнести корпоративную культуру, организации рабочих процессов, предписанные к использованию инструменты разработки, атмосфера внутри компании, методологию разработки и др. Все это может повлиять на мотивацию сотрудника и качество его работы.

Также серьезное влияние может оказывать отсутствие понимания продукта и специфики задач. При оценке важно понимать над чем именно работает разработчик или команда: над исправлением багов или написанием фич. В первом случае требуется большое количество исследований и экспериментов, во втором – скрупулезности и тестирования. В таком случае при оценке одного и того же показателя команда, которая работает с более сложными задачами, может показаться менее продуктивной. Также необходимо учитывать, работает ли команда с легаси-кодом или пишет решение с нуля, используются ли какие-то решения с открытым кодом, так как это тоже влияет на процесс разработки и результаты его оценивания.

Список использованных источников

1. ДеМарко, Т. Контроль за ходом разработки программного обеспечения: управление, измерение и оценка / Т. ДеМарко. – Москва : ДМК Пресс, 2004. – 288 с.
2. Зак Весталл. Измерение продуктивности разработчиков [Электронный ресурс]. – Режим доступа: <https://linearb.io/blog/developer-productivity>. – Дата доступа: 07.03.2025.
3. Кон, М. Agile. Оценка и планирование проектов : учеб.-метод. пособие / М. Кон. – Т. 91 (Библиотека Сбера). – Москва : СберУниверситет, 2021. – 320 с.
4. Лайман, И. Измерение продуктивности разработчиков [Электронный ресурс]. – Режим доступа: <https://stackoverflow.blog/2020/12/07/measuring-developer-productivity/>. – Дата доступа: 07.03.2025.
5. Пейдж Круз. Руководство по измерению продуктивности разработчиков [Электронный ресурс]. – Режим доступа: <https://chronosphere.io/learn/a-starting-guide-to-measuring-developer-productivity/>. – Дата доступа: 07.03.2025.
6. Метрики качества кода, которые следует измерять [Электронный ресурс]. – Режим доступа: <https://www.future-processing.com/blog/code-quality-metrics-that-you-should-measure/>. – Дата доступа: 07.03.2025.