

*Двоеглазов Денис Александрович*¹,
студент, бакалавр;
*Магер Владимир Евстафьевич*²,
канд. техн. наук, ст. науч. сотр., доцент

ОЦЕНКА РЕЗУЛЬТАТИВНОСТИ ОПТИМИЗАЦИИ ПРОЦЕССА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ПРИМЕНЕНИЕМ НЕПРЕРЫВНОЙ ИНТЕГРАЦИИ

^{1,2} Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Россия,
¹ denisdvoeglazov1996@gmail.com, ² mv@qmd.spbstu.ru

Аннотация. В работе произведен анализ применения подхода непрерывной интеграции в целях оптимизации процесса разработки программного обеспечения и выявлены особенности указанного метода. Полученные оценки результативности доказывают эффективность использования непрерывной интеграции.

Ключевые слова: непрерывная интеграция, результативность, разработка программного обеспечения.

*Denis A. Dvoeglazov*¹,
Bachelor Student;
*Vladimir E. Mager*²,
Candidate of Technical Sciences, Associate Professor

PERFORMANCE ASSESSMENT OF THE SOFTWARE DEVELOPMENT PROCESS' OPTIMIZATION USING CONTINUOUS INTEGRATION

^{1,2} Peter the Great St.Petersburg Polytechnic University, St.Petersburg, Russia,
¹ denisdvoeglazov1996@gmail.com, ² mv@qmd.spbstu.ru

Abstract. Analyzes of using of the continuous integration aimed on optimization of the software development process, as well as identifying of peculiarities of this method, are carried out in the paper. The resulting performance assessments prove the effectiveness of using of continuous integration.

Keywords: continuous integration, effectiveness, software development.

Введение. Повышение эффективности процессов, связанных с разработкой программного обеспечения, является востребованной и актуальной задачей. Одним из наиболее перспективных направлений опти-

мизации данных процессов является применение непрерывной интеграции [1].

Непрерывная интеграция (НИ) представляет собой практику разработки программного обеспечения, которая заключается в выполнении частых автоматизированных сборок проекта. Данный подход позволяет своевременно обнаруживать и решать проблемы, возникающие в процессе разработки [2]. Однако использование НИ предоставляет разработчикам программного обеспечения и другие преимущества. Например, в работе [3] описаны возможности средств НИ с точки зрения автоматизации тестирования. Кроме того, использование данного подхода позволяет ускорить ввод программного продукта в эксплуатацию за счет обработки сред и конфигураций [4].

Данная работа посвящена оценке результативности применения непрерывной интеграции в процессе разработки программного обеспечения.

Постановка задачи. Основным критерия результативности процесса разработки программного обеспечения является время t_r , затрачиваемое на этапах написания программного кода, тестирования и ввода в эксплуатацию разрабатываемого продукта. Для оценки результативности применения НИ в процессе разработки программного обеспечения используется соотношение:

$$R = \frac{T}{t_r}, \quad T = \min(t_r, t_p), \quad (1)$$

где t_p – время, отведенное на выполнение соответствующих этапов процесса разработки.

Исходные данные. Для того чтобы оценить результативность применения подхода непрерывной интеграции в процессе разработки программного обеспечения (ПО), были проанализированы 5000 проектов разработчиков с открытым исходным кодом, находящихся в общем доступе в онлайн-сервисе GitHub, и имеющими наибольший рейтинг GitHub за последний месяц.

По выбранным проектам была собрана следующая информация:

- название проекта;
- используется ли НИ;
- средство НИ (для проектов, использующих НИ);
- рейтинг GitHub;
- среднее время релиза (выхода новой версии ПО);
- количество релизов;
- основной язык программирования.

Для сбора данных был написан скрипт на языке Python, использующий API GitHub. Использование подхода непрерывной интеграции опре-

делялось на основе наличия в репозитории проекта конфигурационных файлов вида *-ci.yml.

Оценка результативности. Собранные данные о проектах были проанализированы на предмет использования подхода НИ. Результаты анализа свидетельствуют о том, что разработка ПО более чем в 57 % случаев проводится с использованием средств непрерывной интеграции (см. табл. 1).

Таблица 1

Количество проектов, использующих процессы непрерывной интеграции

Используется ли НИ	Число проектов	Доля проектов, %
Да	2863	57,26
Нет	2137	42,74

Также были определены наиболее популярные средства непрерывной интеграции. В большинстве случаев разработчики ПО отдают предпочтение двум системам – GitLab и Travis CI. При этом значительным преимуществом первой является поддержка тестирования ПО для ОС Linux и Windows, в то время как для Travis CI данная опция доступна только для ОС Linux. В таблице 2 приведен процент использования разработчиками различных средств НИ.

Таблица 2

Использование средств непрерывной интеграции

Показатель	GitLab	Travis CI	CircleCI	Jenkins	CloudBees
Доля проектов, %	44	39	11	6	<1
Число проектов	1259	1116	315	171	2

Для того чтобы определить, зависит ли рейтинг проекта от использования непрерывной интеграции, все проекты были разбиты на 25 групп согласно рейтингу GitHub. Каждая группа содержит около 200 проектов. После чего для каждой группы был вычислен процент проектов, использующих НИ (см. рис. 1).

Полученное распределение позволяет прийти к заключению, что в проектах с высоким рейтингом непрерывная интеграция применяется чаще. Так, в группе проектов с наибольшим рейтингом в 88% случаев при разработке применялась НИ. По мере уменьшения рейтинга проектов снижается и процент проектов, использующих НИ.

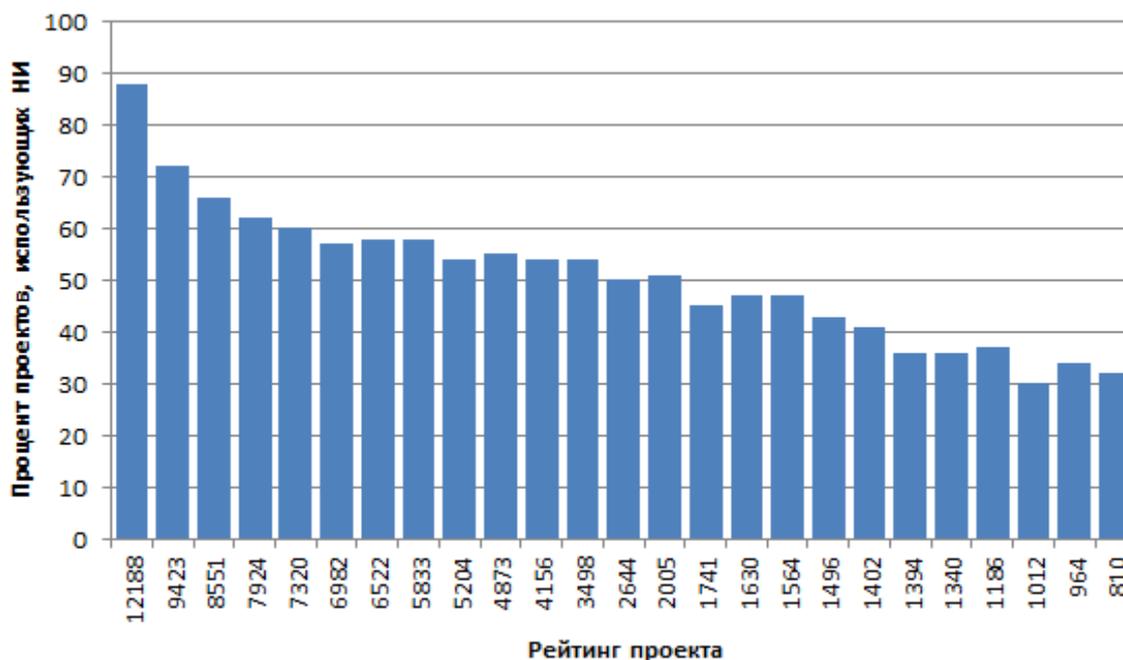


Рис. 1. Зависимость использования средств НИ от рейтинга проекта

Для анализируемых проектов было рассчитано, как часто производилось изменение настроек НИ (см. табл. 3).

Таблица 3

Частота изменения конфигурации средств непрерывной интеграции

Количество изменений конфигурации средств НИ	1	от 1 до 10	от 10 до 100	более 100
Доля проектов, %	2	63	26	9
Число проектов	57	1803	745	258

На основе полученных результатов можно сделать вывод о том, что в большинстве случаев для реализации проекта не требуется частой настройки средств НИ. То есть конфигурация средств НИ не требует больших временных затрат.

Для оценки времени, затрачиваемого на выпуск ПО, была рассчитана частота релизов для каждого из рассматриваемых проектов. Результат приведен в таблице 4.

Таблица 4

Частота выпусков ПО

Используются ли средства НИ	Количество релизов, шт./месяц
ДА	0,68
НЕТ	0,31

Таким образом, использование непрерывной интеграции в процессе разработки ПО позволяет увеличить скорость выпуска ПО в 2,2 раза.

Для оценки результативности применения разработчиками ПО подхода непрерывной интеграции согласно (1) было определено среднее количество времени, затрачиваемое на выпуск ПО (см. табл. 5).

Таблица 5

Среднее время выпуска ПО

Используются ли средства НИ	Среднее время выпуска ПО, дней
ДА	44
НЕТ	97

В качестве времени t_r принято среднее время выпуска последних 20 % релизов проекта, в качестве времени t_p – среднее время для всех релизов проекта.

Среди всех проектов, отобранных для анализа, были выделены двадцать проектов со средним временем релиза, наиболее близким к значениям, приведенным в таблице 5. Для этих проектов была вычислена оценка результативности R использования НИ (см. табл. 6).

Таким образом, результативность процесса разработки ПО с использованием средств непрерывной интеграции превосходит результативность процесса разработки без использования данного подхода в 1,5 раза.

Заключение

На основании проведенного исследования можно сделать вывод о том, что применение принципов непрерывной интеграции оптимизирует процесс разработки программного обеспечения и позволяет существенно сократить время выпуска продукта.

Таблица 6

Оценка результативности

№ проекта	Разработка ПО с применением НИ			Разработка ПО без применения НИ		
	t_r	t_p	R	t_r	t_p	R
1	50	42	0,84	146	95	0,65
2	46	42	0,91	170	95	0,56
3	39	43	1	138	96	0,7
4	47	44	0,94	169	96	0,57
5	55	44	0,8	158	97	0,61
6	51	44	0,86	184	97	0,53
7	49	45	0,92	151	99	0,66
8	42	45	1	172	99	0,58
9	53	45	0,85	160	99	0,62
10	52	46	0,88	167	101	0,6
Среднее значение R	0,9			0,61		

Список литературы

1. Дюваль П.М. Непрерывная интеграция. Улучшение качества программного обеспечения и снижение риска. М.: Вильямс, 2008. 345 с.
2. Кузьмина И.В., Фидельман В.Р. Разработка программного обеспечения сложных аппаратно-программных комплексов с использованием принципов непрерывной интеграции // Известия высших учебных заведений. Поволжский регион. Технические науки. 2012. № 2 (22). С. 44–49.
3. Beller M., Gousios G., Zaidman A. Oops, my tests broke the build: An analysis of travis ci builds with github // PeerJ Preprints. 2016. № e1984v1.
4. Leppänen M., Mäkinen S., Pagels M., Eloranta V.P., Itkonen J., Mäntylä M.V., Männistö T. The highways and country roads to continuous deployment // IEEE software. 2015. № 2 (32). P. 64–72.