

Университет ИТМО, факультет программной инженерии и компьютерной техники
Двухнедельная отчётная работа по «Информатике»: аннотация к статье

Дата прошедшей лекции	Номер прошедшей лекции	Название статьи/главы книги/видеолекции	Дата публикации (не старше 2021 года)	Размер статьи (от 400 слов)	Дата сдачи
11.09.2024	1	Энтропия. Как хаос помогает искать вирусы	29.01.2021	~1300	25.09.2024
25.09.2024	2	Скажи-ка, дядя, ведь не RAR'ом... Почему архивирование данных не менее важно, чем бэкап	16.11.2021	~1400	09.10.2024
09.10.2024	3	Как Rust стал любимым языком программистов	17.02.2023	~1600	23.10.2024
23.10.2024	4	XML схемы в строительстве	16.08.2022	~1200	06.11.2024
06.11.2024	5	Вместо офиса: как отказаться от решений Google и Microsoft на случай сами знаете чего	28.05.2024	~950	20.11.2024
20.11.2024	6	Что такое Pixel Perfect вёрстка и зачем она нужна	21.05.2021	~1050	04.12.2024
04.12.2024	7	The Role of Smoke Tests In Software Development	18.10.2023	~1950	18.12.2024

Выполнил(а) _____, № группы P3115, оценка _____
Фамилия И.О. студента не заполнять

Прямая полная ссылка на источник или сокращённая ссылка (bit.ly, tr.im и т.п.)

<https://www.split.io/blog/smoke-tests-in-software-development/>

Теги, ключевые слова или словосочетания (минимум три слова)

Дымовое тестирование, автоматизированное тестирование, проверка стабильности сборки ПО, жизненный цикл разработки ПО, непрерывная интеграция, флаги функций

Перечень фактов, упомянутых в статье (минимум четыре пункта)

- 1) Дымовое тестирование – это предварительный процесс тестирования, проверяющий стабильность сборки ПО.
- 2) Основная цель дымового тестирования – на ранней стадии найти критические ошибки (без которых продукт не сможет нормально функционировать), при этом не погружаясь в детали.
- 3) Дымовое тестирование чаще всего проводится при выпуске новой сборки или версии ПО.
- 4) Если сборка проходит «дымовые тесты», это означает, что она достаточно стабильна для перехода к более детальному тестированию.
- 5) Благодаря дымовому тестированию предотвращается перевод нестабильных сборок на глубокое тестирование.
- 6) Дымовые тесты обычно проводятся достаточно быстро, что позволяет получать информацию о серьезных проблемах быстрее, чем при проведении более детального тестирования.
- 7) Флаги функций дают возможность включать и выключать определенные функции без необходимости полного развертывания всего приложения.
- 8) Дымовое тестирование позволяет быстро проверять добавленные функции на критические ошибки для всего приложения.

Позитивные следствия и/или достоинства описанной в статье технологии (минимум три пункта)

- 1) Экономия времени и ресурсов за счет быстрого тестирования критических ошибок.
- 2) Частая проверка ключевых функций снижает риски добавления в основную базу кода, нарушающего работу этих функций.
- 3) Возможность поддержки непрерывной интеграции, так как такое тестирование позволяет быстро получать информацию о работоспособности сборки.

Негативные следствия и/или недостатки описанной в статье технологии (минимум три пункта)

- 1) Качество такого тестирования очень зависит от качества написанных тестов (получение позитивного результата не всегда означает, что сборка стабильна).
- 2) Проверка достаточно ограниченного количества функций (можно пропустить важные, но менее очевидные) детали.
- 3) Необходимость постоянного обновления с увеличением функциональности, что может быть довольно трудоемким процессом.
- 4) Дымовые тесты могут давать ложное представление о стабильности проекта, из-за чего может быть пропущен этап детального тестирования.

Ваши замечания, пожелания преподавателю или анекдот о программистах¹