

КРАТКИЕ СООБЩЕНИЯ

BRIEF PAPERS

doi: 10.17586/2226-1494-2024-24-2-330-334

УДК 004.4

Подход к разработке программных продуктов в стартапе

Наталия Владимировна Лонкина¹, Любовь Сергеевна Лисицына²✉

^{1,2} Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация

¹ nataliia.lonkina@gmail.com, <http://orcid.org/0009-0002-2249-3822>

² lisizina@itmo.ru✉, <http://orcid.org/0000-0002-1493-5849>

Аннотация

Исследована проблема срывов сроков релиза в условиях систематического изменения требований рынка к программному продукту. Проведен анализ причин срывов на всех этапах разработки программного продукта и предложен подход, направленный на поиск компромисса между качеством и сроком внедрения разрабатываемого продукта для сокращения времени выхода релиза. Представлены результаты практического применения данного подхода на примере стартапа по разработке сервиса тест-драйва техники, которые подтвердили его эффективность: время релиза сократилось на 15 %.

Ключевые слова

стартап, этапы разработки программных продуктов, причины срыва сроков релиза, способы сокращения сроков разработки программных продуктов

Ссылка для цитирования: Лонкина Н.В., Лисицына Л.С. Подход к разработке программных продуктов в стартапе // Научно-технический вестник информационных технологий, механики и оптики. 2024. Т. 24, № 2. С. 330–334. doi: 10.17586/2226-1494-2024-24-2-330-334

Approach to software products development in a startup

Nataliia V. Lonkina¹, Liubov S. Lisitsyna²✉

^{1,2} ITMO University, Saint Petersburg, 197101, Russian Federation

¹ nataliia.lonkina@gmail.com, <http://orcid.org/0009-0002-2249-3822>

² lisizina@itmo.ru✉, <http://orcid.org/0000-0002-1493-5849>

Abstract

The problem of missed release deadlines under conditions of systematic changes in market requirements for a software product has been studied. The analysis of the causes of disruptions at all stages of software product development is carried out and an approach is proposed aimed at finding a compromise between the quality and the implementation period of the product being developed in order to reduce the release time. The results of the presented method are analyzed using the example of development in a startup at the early stages of development and confirmed the possibility of reducing the time for updating software products by at least 15 %.

Keywords

startup, stages of software product development, reasons for failure of release dates, ways to reduce the time of software product development

For citation: Lonkina N.V., Lisitsyna L.S. Approach to software products development in a startup. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2024, vol. 24, no. 2, pp. 330–334 (in Russian). doi: 10.17586/2226-1494-2024-24-2-330-334

Ежедневно на рынке России появляются сотни новых стартапов¹, среди которых особое место занимают проекты по разработке инновационных программных продуктов (ПП).

В условиях высокой конкуренции крайне важно не просто создать новый продукт, соответствующий актуальным требованиям рынка, но и сделать это в минимальные сроки. Часто характерной особенностью таких стартапов является ограниченность ресурсов. В таких условиях сократить время разработки от идеи до готового продукта можно за счет применения гибких, масштабируемых и легко адаптируемых к изменениям подходов.

В настоящей работе рассмотрен подход к разработке нового ПП на основе методологии Agile [1, 2], главным преимуществом которой является организация процессов разработки таким образом, чтобы максимально упростить внесение изменений в код ПП без значительных трудозатрат и временных потерь [3]. Это позволит не только быстро выводить на рынок новые ПП, но и противостоять конкурентам, оперативно реагируя на запросы его потенциальных потребителей и на различные изменения требований рынка.

На рисунке показаны основные этапы разработки ПП, которые повторяются в цикле, начиная с формирования требований и заканчивая внедрением. Рассмотрим причины, которые могут замедлить выпуск продукта на рынок на каждом этапе и повлечь за собой необходимость изменять или полностью переписывать код ПП, и принципы для их устранения.

На этапе формирования требований к ПП (этап 1) необходимо опираться на потребности рынка. Ошибки и недочеты на этом этапе могут существенно снизить ценность разрабатываемого ПП [4]. Требования к ПП должны быть формализованы и написаны развернуто, подробно и четко, чтобы быть понятными для всех участников разработки.

При планировании работ по разработке ПП (этап 2) следует опираться, прежде всего, на метрику time-to-market [5]. В стартапе данный этап может оказаться проблемным, так как могут возникнуть принципиально новые задачи, которые до этого никто не решал. Один из эффективных приемов планирования быстрого выпуска ПП — ранжирование задач и требований к их решению с возможным исключением задач и/или требований с низким приоритетом [6].

На этапе проектирования ПП (этап 3) предпочтение отдается тем решениям, которые обеспечивают оптимальное сочетание скорости разработки, гибкости и масштабируемости информационной системы (ИС). На этом этапе необходимо создавать такой проект, который позволит не разрабатывать ИС или ее отдельные модули заново.

В условиях сжатых сроков разработки на этапе кодирования (этап 4) часто возникают ошибки в коде. Пишется или создается код с низкой производи-



Рисунок. Основные этапы разработки программных продуктов

Figure. The main stages of software product development

тельностью, который невозможно дополнить в будущем. Исходя из этого, основное внимание в процессе написания кода ПП в стартапе уделяется построению чистого, модульного кода с использованием современных инструментов и фреймворков, способствующих скорости и простоте внесения изменений. Во время этого этапа важно придерживаться принятой архитектуры разрабатываемого программного обеспечения, сверяться с установленными требованиями, тщательно выбирать инструменты кодирования, соблюдать общепринятые стандарты, использовать контроль версий, документировать код и обеспечивать его модульными тестами.

Этап тестирования (этап 5) связан с поиском ошибок, которые не были обнаружены ранее на этапе 4. Важность этого этапа определяется тем, что оставшиеся после него ошибки в коде придется исправлять после внедрения системы, что может сдвинуть сроки релиза и существенно снизить качество продукта. По этой причине, основной акцент уделяется разработке качественных сценариев тестирования ПП.

Завершающий этап разработки ПП — внедрение (этап 6), в результате которого ИС становится доступной пользователям. На данном этапе выполняется подготовка тестового окружения. После развертывания и проверки работоспособности ИС на тестовом окружении происходит ее внедрение на том окружении, которое используют конечные пользователи системы. Крайне важно предусмотреть, чтобы развертывание системы происходило быстро, а сама система была защищена от различных атак и обладала отказоустойчивостью. В случае несоблюдения этих требований может произойти утечка данных, а система может стать для пользователей недоступной. От этого этапа зависит то, насколько безошибочно и быстро начнет свою работу ИС с реальными пользователями.

На каждом из описанных этапов разработки ПП существуют угрозы, которые могут стать препятствиями для своевременного релиза или оказать влияние на его дальнейшее развитие (табл. 1).

Разработка ПП в стартапе — это всегда некий компромисс между полнотой функциональности и сроком выпуска (обновления) продукта. Часто приходится стремиться не к совершенству, а к запуску новой версии продукта в срок, чтобы затем получать отзывы его пользователей и на их основе улучшать такой продукт [2]. В связи с этим при разработке ПП в стартапе важно уметь расставлять приоритеты и фокусироваться на

¹ Клейменова Л. От идеи до единорога — стартапы России и мира в 22 цифрах [Электронный ресурс]. Режим доступа: <https://trends.rbc.ru/trends/innovation/5f04aeac9a79479c0727f494> (дата обращения: 07.02.2024).

Таблица 1. Причины угроз срыва сроков релиза
Table 1. The reasons for the threats of disruption of product release dates

Номер угрозы	Причины угроз	Этапы разработки программных продуктов					
		1	2	3	4	5	6
1	Добавление нового функционала	+	–	+	+	–	+
2	Изменение существующего функционала	+	+	+	+	–	+
3	Отказ от существующего функционала	+	+	+	+	–	–
4	Наличие ошибок в коде	–	–	–	+	+	–
5	Недостаточная производительность ИС	–	–	–	+	+	–

Примечание. «+» — присутствие угрозы на этапе; «–» — отсутствие угрозы.

главном. Сгруппируем угрозы, выделенные в табл. 2, и опишем способы их предотвращения.

Группа угроз 1 связана с добавлением, изменением или отказом от функционала, которые выявлены в актуальных исследованиях рынка. Неопределенность требований, необходимость релиза в запланированное время, ограниченность ресурсов — все это делает эти угрозы широко распространенными в стартапе.

При составлении требований важно понимать, какой именно продукт нужно развивать, как выполнить все требования, и как своевременно и корректно донести информацию о требованиях до участников разработки ПП. Для этого следует использовать фиксацию всех требований в вики или трекере задач с расстановкой приоритетов (необходимо помнить, что в случае нарушения сроков задачи с низким приоритетом в план разработки могут не попасть), а также заранее создавать и согласовывать макет ПП. Данные действия могут уменьшить количество разногласий между разработчиками.

При планировании нужно не просто расставлять приоритеты задач, но и делать оценку их трудоемкости, которая часто бывает далекой от реальной (особенно тогда, когда разработчики ни разу не решали подобные задачи). Рекомендуется не только использовать различные методы оценки, но и внедрять в работу сторипонты, и использовать экспертные оценки.

При проектировании системы следует закладывать фундамент для ее масштабирования и гибкости к изменениям [7], стремиться к соответствию принци-

пам SOLID (Single responsibility, Open-closed, Liskov substitution, Interface segregation и Dependency inversion) [8]. В результате важно разбить систему на независимые компоненты, каждый из которых имеет единственную зону ответственности. Тогда при неудачной реализации этапа 4 можно будет обновить систему поэтапно без приостановки ее работы. При возможности следует использовать уже готовые компоненты, например, развернутые хранилища данных. Система должна разворачиваться быстро с соблюдением требования непрерывной интеграции и развертывания на серверах, что обеспечивается за счет автоматизации этих работ. Для обеспечения безопасности системы здесь следует спрятать секреты, настроить зеркалирование, предусмотреть защиту от DDOS-атак и т. п.

Группа угроз 2 связана с наличием ошибок в коде и с недостаточностью производительности разрабатываемой ИС. Разработка новых продуктов в сжатые сроки в стартапе часто влечет за собой большое количество ошибок и недоработок в коде и, соответственно, его низкую производительность. Как правило, в таких случаях программистам нужно исправлять ошибки и применять новые подходы для повышения производительности, однако, в случае разработки продуктов в сжатые сроки, это не всегда возможно.

Рассмотрим способы устранения таких угроз.

При кодировании следует писать код так, чтобы его было максимально легко понять, исправить и использовать повторно. При этом особое внимание следует уделять выбору инструментов разработки. Язык про-

Таблица 2. Способы предотвращения угроз срыва сроков релиза
Table 2. Ways to prevent threats of disruption of the release dates of software product

Номер способа	Способ	Угрозы				
		1	2	3	4	5
1	Фиксация задач в вики или трекере с расстановкой их приоритетов	+	+	+	–	–
2	Использование различных методов оценки задач	+	+	+	–	–
3	Разбиение системы на компоненты с единой зоной ответственности	+	+	+	–	–
4	Использование готовых компонентов	+	+	+	–	–
5	Автоматизация поставки и развертывания системы	+	+	+	–	–
6	Использование контроля версий	–	–	–	+	+
7	Покрывание кода тестами, исправление только критических дефектов	–	–	–	+	+

Примечание. «+» — возможность способа предотвращения угроз; «–» — отсутствие угроз.

Таблица 3. Принятые решения и результаты их реализации в стартапе Lampu
 Table 3. The decisions made and the results of their implementation in the Lampu startup

Номер способа	Реализованное решение	Номер итерации	Трудоемкость без использования предложенных решений, ч	Сокращение трудоемкости за счет предложенного решения, ч
2	Декомпозиция задач с использованием различных методов их оценки	2–6	130	20
3	Выделение независимых модулей с единой зоной ответственности	3–6	150	40
4	Использование готовых компонентов	4–6	140	30
7	Исправление только критических дефектов кода	5–6	117	7
5	Использование CI/CD pipeline на платформе Gitlab для автоматизации развертывания системы	6	113	3
Итого:			650	100

граммирования, библиотеки — все это должно быть знакомо разработчикам и предоставлять возможность писать простой код [9]. Кроме того, при кодировании необходимо вести контроль версий и тестировать написанный код, что позволит сделать его разработку более удобной, качественной и быстрой.

При тестировании важен процесс передачи дефектов разработчикам. Имеет большое значение определение дефектов ПП в таск-менеджере и, аналогично требованиям, проставить уровень их критичности (важности). Для сокращения времени выпуска ПП следует в первую очередь закрывать критичные дефекты, а остальные вносить в бэклог.

Опишем результаты практического применения предложенного подхода в стартапе Lampu. В стартапе разработана ИС, которая является, с одной стороны, агрегатором, а с другой — магазином товаров, и состоит из следующих модулей: фронтенд, бэкенд, реляционная база данных, нереляционная база данных, сервер переадресации. С октября 2023 года в разработке ИС приняло участие четыре программиста, два из которых являлись старшими инженерами, а два — стажерами из числа студентов Университета ИТМО.

В рамках эксперимента по преодолению угроз срыва сроков релиза (табл. 1) были использованы 6 коротких итераций [10] для разработок: 1) главной страницы веб-приложения¹; 2) страницы каталога²; 3) страницы

товаров³; 4) корзины⁴; 5) страницы для аутентификации, авторизации, регистрации⁵; 6) страницы для оформления заказа⁶. Итерация 1 проведена с использованием способов 1 и 6, а на итерациях 2–6 проведены эксперименты, в которых измерено время разработки на каждой итерации с и без применения решений. Далее обновление ПП выполнено с применением перечисленных способов (табл. 2). В табл. 3 представлены результаты реализации предложенных решений: общая трудоемкость разработки кода продукта без использования таких решений составила 650 ч, ее удалось сократить примерно на 100 ч, что составляет 15 %.

Изложенный в работе подход к разработке ПП, а также его практическая реализация в стартапе Lampu может быть полезным для создателей новых ИС и помогут им минимизировать сроки релизов в ответ на актуальные вызовы рынка. Среди предложенных решений данного подхода следует отметить следующие реализованные решения: использование различных методов оценки задач; разбиение системы на компоненты с единственной зоной ответственности; использование сторонних готовых компонентов; автоматизацию поставки и развертывания системы; исправление только критичных дефектов кода.

³ [Электронный ресурс]. Режим доступа: <https://lampu.store/product-list/smartphones> (дата обращения: 07.02.2024).

⁴ [Электронный ресурс]. Режим доступа: <https://lampu.store/cart> (дата обращения: 07.02.2024).

⁵ [Электронный ресурс]. Режим доступа: <https://lampu.store/authentication> (дата обращения: 07.02.2024).

⁶ [Электронный ресурс]. Режим доступа: <https://lampu.store/gocheckout> (дата обращения: 07.02.2024).

¹ [Электронный ресурс]. Режим доступа: <https://lampu.store> (Сервис тест-драйва техники) (дата обращения: 07.02.2024).

² [Электронный ресурс]. Режим доступа: <https://lampu.store/catalog> (дата обращения: 07.02.2024).

Литература

- McCaffery F., Taylor P.S., Coleman G. Adept: A unified assessment method for small software companies // *IEEE Software*. 2007. V. 24. N 1. P. 24–31. <https://doi.org/10.1109/ms.2007.3>
- Tegegne E.W., Seppänen P., Ahmad M.O. Software development methodologies and practices in start-ups // *IET Software*. 2019. V. 13. N 6. P. 497–509. <https://doi.org/10.1049/iet-sen.2018.5270>

References

- McCaffery F., Taylor P.S., Coleman G. Adept: A unified assessment method for small software companies. *IEEE Software*, 2007, vol. 24, no. 1, pp. 24–31. <https://doi.org/10.1109/ms.2007.3>
- Tegegne E.W., Seppänen P., Ahmad M.O. Software development methodologies and practices in start-ups. *IET Software*, 2019, vol. 13, no. 6, pp. 497–509. <https://doi.org/10.1049/iet-sen.2018.5270>

3. Highsmith J., Cockburn A. Agile software development: The business of innovation // *Computer*. 2001. V. 34. N 9. P. 120–127. <https://doi.org/10.1109/2.947100>
4. Mullins J.W., Sutherland D.J. New product development in rapidly changing markets: An exploratory study // *Journal of Product Innovation Management*. 1998. V. 15. N 3. P. 224–236. <https://doi.org/10.1111/1540-5885.1530224>
5. Sawyer P., Sommerville I., Kotonya G. Improving market-driven re processes // *Proc. of the International Conference on Product Focused Software Process Improvement*. Oulu, Finland, 1999. P. 222–236.
6. Karlsson L., Dahlstedt Å.G., Regnell B., Natt och Dag J., Persson A. Requirements engineering challenges in market-driven software development – an interview study with practitioners // *Information and Software Technology*. 2007. V. 49. N 6. P. 588–604. <https://doi.org/10.1016/j.infsof.2007.02.008>
7. Yoffie D.B., Cusumano M.A. Building a company on Internet time: lessons from Netscape // *California Management Review*. 1999. V. 41. N 3. P. 8–28. <https://doi.org/10.2307/41165995>
8. Madasu V.K., Venna T.V.S.N., Eltaieb T. SOLID Principles in Software Architecture and Introduction to RESM Concept in OOP // *Journal of Multidisciplinary Engineering Science and Technology*. 2015. V. 2. N 2. P. 1–3.
9. Giardino C., Unterkalmsteiner M., Paternoster N., Gorschek T., Abrahamsson P. What do we know about software development in startups? // *IEEE Software*. 2014. V. 31. N 5. P. 28–32. <https://doi.org/10.1109/ms.2014.129>
10. Ruparelia N.B. Software development lifecycle models // *ACM SIGSOFT Software Engineering Notes*. 2010. V. 35. N 3. P. 8–13. <https://doi.org/10.1145/1764810.1764814>
3. Highsmith J., Cockburn A. Agile software development: The business of innovation. *Computer*, 2001, vol. 34, no. 9, pp. 120–127. <https://doi.org/10.1109/2.947100>
4. Mullins J.W., Sutherland D.J. New product development in rapidly changing markets: An exploratory study. *Journal of Product Innovation Management*, 1998, vol. 15, no. 3, pp. 224–236. <https://doi.org/10.1111/1540-5885.1530224>
5. Sawyer P., Sommerville I., Kotonya G. Improving market-driven re processes. *Proc. of the International Conference on Product Focused Software Process Improvement*. Oulu, Finland, 1999, pp. 222–236.
6. Karlsson L., Dahlstedt Å.G., Regnell B., Natt och Dag J., Persson A. Requirements engineering challenges in market-driven software development — an interview study with practitioners. *Information and Software Technology*, 2007, vol. 49, no. 6, pp. 588–604. <https://doi.org/10.1016/j.infsof.2007.02.008>
7. Yoffie D.B., Cusumano M.A. Building a company on Internet time: lessons from Netscape. *California Management Review*, 1999, vol. 41, no. 3, pp. 8–28. <https://doi.org/10.2307/41165995>
8. Madasu V.K., Venna T.V.S.N., Eltaieb T. SOLID Principles in Software Architecture and Introduction to RESM Concept in OOP. *Journal of Multidisciplinary Engineering Science and Technology*, 2015, vol. 2, no. 2, pp. 1–3.
9. Giardino C., Unterkalmsteiner M., Paternoster N., Gorschek T., Abrahamsson P. What do we know about software development in startups? *IEEE Software*, 2014, vol. 31, no. 5, pp. 28–32. <https://doi.org/10.1109/ms.2014.129>
10. Ruparelia N.B. Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, 2010, vol. 35, no. 3, pp. 8–13. <https://doi.org/10.1145/1764810.1764814>

Авторы

Лонкина Наталья Владимировна — студент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, <http://orcid.org/0009-0002-2249-3822>, nataliia.lonkina@gmail.com

Лисицына Любовь Сергеевна — доктор технических наук, профессор, профессор, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, [sc 56203524600](http://orcid.org/0000-0002-1493-5849), <http://orcid.org/0000-0002-1493-5849>, lisizina@itmo.ru

Authors

Nataliia V. Lonkina — Student, ITMO University, Saint Petersburg, 197101, Russian Federation, <http://orcid.org/0009-0002-2249-3822>, nataliia.lonkina@gmail.com

Liubov S. Lisitsyna — D.Sc., Full Professor, ITMO University, Saint Petersburg, 197101, Russian Federation, [sc 56203524600](http://orcid.org/0000-0002-1493-5849), <http://orcid.org/0000-0002-1493-5849>, lisizina@itmo.ru

Статья поступила в редакцию 16.02.2024

Одобрена после рецензирования 01.03.2024

Принята к печати 22.03.2024

Received 16.02.2024

Approved after reviewing 01.03.2024

Accepted 22.03.2024



Работа доступна по лицензии
Creative Commons
«Attribution-NonCommercial»