

УДК 338.24

А.И. ЮДИНА,
магистрант

Казанский (Приволжский) федеральный университет

А.Р. САФИУЛЛИН,

доктор экономических наук, профессор

Казанский (Приволжский) федеральный университет

БЕРЕЖЛИВОЕ ПРОИЗВОДСТВО В РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Аннотация. Актуальность применения инструментов бережливого производства в разработке программного обеспечения заключается в том, что компании стремятся принести максимальную пользу пользователям программного обеспечения, употребляя при этом минимальные ресурсы компании-производителя. Благодаря инструментам бережливого производства процесс производства программного обеспечения (далее – ПО) можно сделать удобным для участников команды разработки и сэкономить трудозатраты, используемые при разработке ПО. Целью бережливого производства является процесс производства продукции, построенный таким образом, чтобы потери были минимизированы, насколько это возможно. Для достижения данной цели необходимо добиться эффекта, когда на первом месте будет программное обеспечение как средство для решения бизнес-задач. Инструменты бережливого производства можно применить в процессах производства ПО, построенных по разным методологиям ведения проекта. Например, гибкие методологии ведения проектов нацелены на получение наиболее востребованного продукта на рынке, что говорит о том, что изначально полученные требования от стейкхолдеров могут меняться. Это говорит о том, что при изменении требований от стейкхолдеров может возникнуть потеря – перепроизводство. Данный вид потери можно уменьшить с помощью инструментов бережливого производства, сократив трудозатраты на разработку ПО.

Ключевые слова: оптимизация производства, бережливое производство и ИТ, Lean IT.

Введение

Впервые инструменты бережливого производства были применены в промышленных компаниях. Основоположателем концепции бережливого производства является Тайити Оно, который внедрил данную концепцию в компании Toyota. После этого концепцию бережливого производства начали активно внедрять в компаниях медицинской и банковской сферы. Далее бережливое производство масштабировали для компаний ИТ-сферы. Отсюда и появился термин Lean IT [1]. Одними из первых данную концепцию внедрили компании Motorola, TransUnion и Fujitsu Services.

Бережливое производство в разработке программного обеспечения

В классической концепции бережливого производства выделяют семь видов потерь [2]:

- потери из-за перепроизводства;

- потери времени из-за ожидания;
- потери при ненужной транспортировке;
- потери из-за лишних этапов обработки;
- потери из-за лишних запасов;
- потери из-за ненужных перемещений;
- потери из-за выпуска дефектной продукции.

Как и в других сферах, в ИТ-сфере основной задачей бережливого производства является устранение потерь [3]. В ИТ-сфере потерями являются трудозатраты, не приносящие ценности разрабатываемому продукту в конечном виде.

Концепцию бережливого производства впервые в процессе разработки ПО применили программисты Мэри Поппендик и Том Поппендик. Именно они оптимизировали бережливое производство для ИТ и отразили это в книге «Бережливое производство программного обеспечения». Сами авторы выявили следующие виды потерь [4]:

- простой из-за ожидания;
- задачи, не доведенные до конца;
- функциональные возможности, не требуемые стейкхолдерами;
- частое переключение на другие задачи;
- повторное изучение;
- дефекты функциональных возможностей.

Для устранения этих потерь были разработаны следующие принципы, которые можно легко адаптировать под требования сферы ИТ-услуг [5, 6]:

1. Ликвидировать потери. Данный принцип нацелен на устранение узких мест с неоправданными потерями ресурсов. Для выявления потерь необходимо провести анализ процесса производства и выявить узкие места.

2. Встроить качество. Данный принцип предназначен для получения представления о качестве всей системы. Для этого необходимо проводить проверку качества системы на каждом этапе разработки, что позволяет выявлять ошибки на раннем сроке. В случае упущения данного действия повышается риск увеличения времени на исправление всех ошибок после полной разработки ПО.

3. Создать базу знаний. Данный принцип говорит о том, что для получения необходимой информации и быстрого погружения в работу по проекту необходимо использовать базу знаний. В базе знаний необходимо фиксировать всю информацию по разрабатываемому программному обеспечению и вовремя ее актуализировать.

4. Детально планировать решения. Данный принцип говорит о том, что перед тем, как принимать решения, после исполнения которых будет невозможно вернуться назад, необходимо проанализировать все участки, на которые может повлиять то или иное решение. Это позволит избежать ошибок, которые будет либо невозможно, либо крайне сложно исправить.

5. Откликаться за минимальное время. Данный принцип говорит о том, что необходимо минимизировать время, затрачиваемое на ожидание отклика. Это может быть, например, ожидание при согласовании документов, передаче данных.

6. Уважать людей. Одним из самых важных принципов является уважительное отношение

к членам команды. Именно участники команды могут помочь выявить узкие места в процессе разработки ПО и предложить полезные идеи. Данный принцип эффективен из-за того, что именно сотрудники владеют наибольшим количеством информации о процессе разработки ПО.

7. Помнить о цели. Данный принцип говорит о том, что необходимо при принятии каких-либо решений держать в поле зрения желаемую конечную цель. Это позволит не отклоняться от цели, поставленной в начале проекта.

В бережливом производстве потери могут быть связаны как с материальными ресурсами, так и с сотрудниками, принимающими участие в процессе разработки продукта [7]. Исходя из этого, к потерям можно отнести в том числе и нереализованный потенциал сотрудников, используя который можно снизить трудозатраты или улучшить процесс производства [8].

Внедрению бережливого производства в процесс создания ПО помогут следующие правила:

1. Управление базой знаний. Необходимо всегда иметь актуальную информацию о проекте и разрабатываемом ПО. Многие компании относятся к актуализации информации не так серьезно, как нужно, что приводит к необходимости повторного ее изучения при подключении к проекту новых сотрудников, разработке большого ПО, сдаче отчетной документации заказчикам и т. д. [9].

2. Анализ первопричин. При анализе любых ошибок необходимо стараться найти первопричину ошибки, чтобы решить ее. Это позволит в дальнейшем не повторять подобных ошибок [10] и не тратить дополнительно время на их исправление разными обходными способами.

3. Масштабирование ИТ-системы. При росте компании необходимо увеличивать и мощность ИТ-службы. Это необходимо для своевременного реагирования на возникающие при использовании ПО проблемы. Увеличить мощность ИТ-службы можно как за счет увеличения количества сотрудников и необходимого оборудования, так и оптимизации существующих процессов путем минимизации потерь [11].

Литература

1. Павлов Е.О., Монахов В.А. Производственные системы: теоретические аспекты функционального анализа и перспективы инновационного развития // Российское предпринимательство. – 2018. – Т. 19. – № 10. – С. 3029–3040.
2. Вумек Д.П., Джонс Д.Т. Бережливое производство. Как избавиться от потерь и добиться процветания вашей компании / пер. с англ. – М.: Альпина Паблишер, 2011. – 472 с.
3. Голохтеев К., Матвеев И. Управление производством: инструменты, которые работают. – СПб.: Питер, 2008. – 251 с.
4. Поппендик М., Поппендик Т. Бережливое производство программного обеспечения: от идеи до прибыли / пер. с англ. – М.: Вильямс, 2010. – 256 с.
5. Ясухиро М. Система менеджмента Тойоты / пер. с англ. – М.: Издательство Института комплексных стратегических исследований, 2007. – 198 с.
6. Лайкер Д. Дао Тойота: 14 принципов менеджмента ведущей компании мира. – М.: Альпина Паблишер, 2011. – 205 с.
7. Valitov Sh.M., Sirazetdinova A.Z. Project risks' management model on an industrial enterprise // Asian Social Science. – 2014. – No. 10 (21). – P. 242–249.
8. Petrakis P., Valsamis D. Entrepreneurship, Transaction Costs and Cultural Background // International Business Research. – 2013. – No. 6. – P. 134–138.
9. Jadil Y., Rana N.P., Dwivedi Y.K. A meta-analysis of the UTAUT model in the mobile banking literature: the moderating role of sample size and culture // Journal of Business Research. – 2021. – No. 132. – P. 354–372.
10. Bu F., Wang N., Jiang B., Jiang Q. International journal of information management motivating information system engineers' acceptance of privacy by design in China: an extended UTAUT model // International journal of information management. – 2021. – No. 60. – Art. 102358.
11. Podgornaya A.I., Grudina S.I., Avdonina S.G. An Enterprise Flexible Development Model // Procedia Economics and Finance. – 2015. – Vol. 24. – P. 519–522.

Информация об авторах

Юдина Альбина Ильдаровна, бакалавр, Казанский (Приволжский) федеральный университет.

E-mail: tai21061997@yandex.ru

Сафиуллин Азат Рашитович, доктор экономических наук, профессор, Казанский (Приволжский) федеральный университет.

E-mail: safiullin.ar@gmail.com

A.I. YUDINA,

Master's student

Kazan (Volga region) Federal University

A.R. SAFIULLIN,

Doctor in Economics, Associate Professor

Kazan (Volga region) Federal University

LEAN PRODUCTION IN DEVELOPMENT SOFTWARE

Abstract. The relevance of using lean manufacturing applications in the development of software programs is that the company implements the maximum application for consumers using the minimum resources of the manufacturing company. With the Lean Manufacturing tool, the software production process can be implemented for use in system development. The goal of lean manufacturing is the process of manufacturing products in such a way that waste is minimized as much as possible. For this purpose, it is necessary to achieve a result, where the product will be in the first place as a means of achieving a solution to a business problem. Lean manufacturing tools can be used in software production processes built according to different project research methodologies. For example, agile research methodologies for projects based on the most demanded product on the market suggest that requirements can be obtained from stakeholders. This suggests that the freedom of freedom from stakeholders

can lead to significant loss – overproduction. This type of waste can be reduced by using lean manufacturing tools and reducing labor costs for software development.

Keywords: lean production, lean, production optimization, lean production and IT, lean IT.

References

1. Pavlov E.O., Monakhov V.A. Production Systems: Theoretical Aspects of Functional Analysis and Prospects for Innovative Development // Russian Journal of Entrepreneurship. – 2018. – Vol. 19. – No. 10. – P. 3029–3040.
2. Womack James P., Jones Daniel T. Lean production. How to get rid of losses and achieve prosperity for your company. – M.: Alpina Publisher, 2011.
3. Golokteev K., Matveev I. Production management: tools that work. – St. Petersburg: Peter, 2008.
4. Poppendyk M., Poppendyk T. Lean Software Manufacturing. – Williams / Dialectics, 2010. – P. 100–156.
5. Yasuhiro Modern Toyota management system. – M. Publishing house IKSI, 2007.
6. Liker Jeffrey. The Tao of Toyota: 14 Management Principles for the World's Leading Company. – Moscow: Alpina Publisher, 2011.
7. Valitov Sh.M., Sisazetdinova A.Z. Project risks' management model on an industrial enterprise // Asian Social Science. – 2014. – 10(21). – P. 242–249.
8. Petrakis P., Valsamis D. Entrepreneurship, Transaction Costs and Cultural Background // International Business Research. – 2013. – No. 6. – P. 134–138.
9. Jadal Y., Rana N.P., Dwivedi Y.K. A meta-analysis of the UTAUT model in the mobile banking literature: the moderating role of sample size and culture // J. Bus. Res. – No. 132. – 2021. – P. 354–372.
10. Bu F., Wang N., Jiang B., Jiang Q. International journal of information management motivating information system engineers' acceptance of privacy by design in China: an extended UTAUT model // Int. J. Inf. Manag. – No. 60. – April, 2021. – Article 102358.
11. Podgornaya A.I., Grudina S.I., Avdonina S.G. An Enterprise Flexible Development Model // Procedia Economics and Finance. – Vol. 24. – 2015. – P. 519–522.