

## ПРОГРАММНОЕ СРЕДСТВО АВТОМАТИЗАЦИИ РАСПРЕДЕЛЕНИЯ ЗАДАЧ

Альберт К.Ю.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Куликов С.С. – к.т.н., доцент

В данной статье рассматривается проблематика автоматизации рутинных задач в процессе разработки программного обеспечения, предоставлен алгоритм решения таких задач и архитектура возможного ПО.

Вопросы автоматизации распределения задач всегда являлись актуальными, например, составление расписания в учебном заведении, графика работы станка на предприятии или расписания смен работников общественного питания. Большое количество разработок появилось в середине и в конце 70-х годов прошлого столетия, когда в системах управления стали использовать ЭВМ. В настоящее время вместе со стремительным развитием информационных технологий возникла проблема, как эффективно управлять проектами разработки ПО.

Наиболее распространённая методология управления проектами, которая применяется в настоящий момент – Agile.

Одна из методик Agile — это методика Scrum гибкого управления проектами, помогающая командам структурировать работу и управлять ею на основе набора ценностей, принципов и практик. Методику Scrum чаще всего применяют команды разработчиков приложений, но принципы и опыт её использования можно применить к командной работе любого рода. Это одна из причин такой популярности данной методики.

Главная задача программного средства – оказание помощи разработчикам ПО в автоматизации процесса распределения задач, а т.к. наиболее распространённая методика в разработке ПО – Scrum, то возможность иметь поддержку Atlassian – Jira, которая является наиболее распространённым продуктом для управления задачами, крайне выгодна.

Основополагающая цель в распределении задач – это планирование спринта (спринт – название временного промежутка, за который выполняется определённое количество задач). Компания Atlassian заявляет, что не важна точная планировка задач, так как это не рационально, а нужен примерный план. Составление плана – цель программного средства.

Любая задача оценивается согласно двум основным критериям: приоритетности и сложности, остальные факторы вторичны. На основе этих критериев задачи можно распределять. Если подойти с математической точки зрения, то это классическая задача «NP-полная задача комбинаторной оптимизации», а именно задача о рюкзаке. Однако, задача о рюкзаке не имеет решения за полиномиальное время, а лишь решение перебором (или улучшенным перебором), поэтому вопрос о самом эффективном решении о распределении задач всегда будет открыт.

Для решения данной задачи может быть предложен следующий алгоритм:

- 1) Для входных параметров отсортируем задачи вначале по приоритету;
- 2) Затем отсортируем их по сложности;
- 3) Выберем задачи последовательно и сформируем спринты.

На втором этапе задачи уже отсортированы в том порядке, как если бы один человек выполнял их последовательно без ограничений по времени. Однако, нам нужно распределить задачи по спринтам, и предложенный алгоритм нам полностью подходит. В общем случае задача с большим приоритетом, очевидно, более важна, чем задача с меньшим приоритетом. Задача с одинаковым приоритетом, но большей сложностью, также более важна, чем задача с меньшей сложностью, т.к. при выполнении задачи с большей сложностью может возникнуть много проблемных ситуаций, поэтому её стоит выполнить в первую очередь. Если наша задача не может быть выполнена в течение спринта, то мы её пропускаем и переходим к следующей.

Программное средство следует выполнять на основе микро-сервисной архитектуры. Главным преимуществом является то, что различные модули программного средства для автоматического распределения задач могут разрабатываться и поставляться независимо, что несомненно удобно, поэтому программное средство, которое планируется для автоматизации работы разработчиками, может быть легко усовершенствовано, например, для составления расписания ВУЗа.

### Список использованных источников:

1. Collier, Ken W. (2011). *Agile Analytics: A Value-Driven Approach to Business Intelligence and Data Warehousing*. Pearson Education. pp. 121 ff. ISBN 9780321669544. "What is a self-organizing team".
2. "Microservice architectures: more than the sum of their parts?". IONOS Digitalguide. Retrieved 2022-03-29.
3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*, Second Edition. MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7. Section 34.2: Polynomial-time verification, pp. 979–983.