

---

**А. Д. БУЛОВА, Г. Н. ПОДГОРНАЯ**

---

## **ОЦЕНКА ПРОДОЛЖИТЕЛЬНОСТИ ВЫПОЛНЕНИЯ ИТ-ПРОЕКТА**

---

---

В статье рассматриваются средства (инструменты), применяемые для расчета (прогнозирования) продолжительности выполнения проекта по созданию программного продукта. В зависимости от сложности проекта представлены рекомендации по их применению.

**Ключевые слова:** UseCases; User Story; Backlog; Function Point Analysis.

**УДК** 004.2

---

Оценка продолжительности проекта — это процесс оценки количества времени, которое потребуется для его выполнения. Продолжительность проекта является важным фактором при планировании и составлении графика проекта и часто используется для определения сроков и бюджета проекта. Точные оценки продолжительности проекта могут иметь определяющее влияние на успех проекта, а неточные оценки — привести к задержкам, перерасходу средств и другим нежелательным проблемам.

Знание предполагаемой продолжительности выполнения ИТ-проекта необходимо для успешного планирования и управления данным проектом. Инструменты, такие как UseCases, User Story, Backlog и Function Point Analysis, могут помочь руководителям более точно прогнозировать сроки выполнения проекта. UseCases и User Story помогают определить необходимые задачи и действия, в то время как Backlog и Function Point Analysis позволяют оценить усилия, необходимые для выполнения этих задач. В зависимости от сложности проекта выбираются различные инструменты, поскольку они обеспечивают разный уровень детализации. В связи с этим важно учитывать различные доступные инструменты и выбирать лучший для проекта, чтобы оптимизировать время и ресурсы.

Целью исследования является анализ и обоснование выбора методов прогнозирования продолжительности выполнения проекта, что является важным показателем как для Исполнителя, так и для Заказчика ИТ-проекта.

Для достижения названной цели были поставлены и решены следующие задачи:

показаны пути решения задачи прогнозирования продолжительности выполнения проекта. В зависимости от полученного результата руководством принимается решение — следует выполнять данный проект или нет;

приведен пример применения рассматриваемых подходов.

Как известно [1] существуют три основные характеристики процесса выполнения проекта: цель, бюджет, сроки.

Как определиться со сроками выполнения проекта или во что это выльется по времени?

---

*Александр Дмитриевич БУЛОВА (bulovaad@belhard.com), кандидат технических наук, доцент кафедры экономической информатики Белорусского государственного экономического университета (г. Минск, Беларусь);*

*Галина Николаевна ПОДГОРНАЯ (galpodgornaya@gmail.com), ассистент кафедры экономической информатики Белорусского государственного экономического университета (г. Минск, Беларусь).*

---

Для получения ответа на этот вопрос существуют следующие инструменты:  
1) *пользовательские сценарии* (UserCases). Они построены на основе применения пользовательских историй (Story mapping).

В современных подходах к управлению программными проектами Agile/Scrum [2] применяется развитие идеи пользовательских сценариев, так называемые пользовательские истории (User Story). Пользовательские истории собираются в бэклог (Backlog). Суть записи пользовательских историй следующая — вся функциональность программного продукта разбивается на отдельные (максимально независимые) кусочки. В Scrum их называют «кусочки требований» — *piece of requirement*. И каждый такой «кусочек» записывается в формате:

Я как [пользователь] хочу [действие], чтобы была [польза].

Пользователь — любой пользователь в системе. В несложных системах обычно вводят 3—5 типов пользователей.

Действие — описание функциональности.

Польза — что получает пользователь в результате этого действия. Здесь фиксируется некая бизнес-ценность — ради чего это действие необходимо делать. Рассмотрим примеры пользовательских историй.

<i>Описание функциональности</i>	<i>Пользовательские истории</i>
Электронная библиотека	Я как читатель хочу видеть перечень книг, упорядоченных по авторам, разделам
Система управления обучением. Информирование участника тренинга о месте, дате и стоимости обучения	Я как участник тренинга хочу получить письмо с расписанием занятий и банковские реквизиты на оплату обучения

Подобный подход понятен бизнес-пользователям и может быть структурирован в управлении требованиями пользователей. Пользовательские истории традиционно фиксировались в Scrum в виде карточек или стикеров. Такие карточки называются «пользовательские карточки» — User Card. Однако подобный формат хоть и удобен для команд, которые находятся в одном помещении, он не применим для распределенных команд. В этом случае используются online инструменты. Самые популярные: Jira&Confluence, Google Docs + Spreadsheet, MS Excel, Backlog. Все пользовательские истории собираются в бэклог. Каждая пользовательская история в бэклоге должна содержать: описание; приоритет (обычно чем выше висит карточка с историей, тем выше приоритет); объем (оценка этого требования); релиз/итерация (когда по плану предполагается разработка истории). Последние два требования появляются после планирования проекта.

После того как в проекте получено структурированное дерево работ (СДР, WBS — work breakdown structure), следует перейти к оценке пакетов работ.

При этом разделяют оценку требований, трудозатрат, ресурсов, бюджета.

Оценка проекта проходит в следующем порядке: SCOPE ESTIMATION (оценка области применения) →EFFORT ESTIMATION (оценка трудозатрат) →SCHEDULE ESTIMATION (оценка сроков) →RISK ESTIMATION (оценка рисков) →RESOURCE ESTIMATION (оценка ресурсов) →BUDGET ESTIMATION (оценка бюджета).

Для оценки объема функциональности, размера проекта, сроков, объема работ используются различные методы и подходы. Некоторые подходы лучше применимы к большим (Б) проектам, а некоторые — только к средним (С) и малым (М).

В табл. 1 представлены методы оценки, используемые в программных проектах, и их применимость.

Таблица 1. Методы оценки в программных проектах

Показатель	Функциональная точка	Story Points	Метод футболки	Метод аналогий	Экспертная оценка	Метод Дельфи
Что оценивается		Размер, объем работ, сроки, функциональность	Размер, объем работ, сроки, функциональность	—	—	—
Размер проекта		М С Б	С Б	—	—	—
Стадия разработки		Ранняя-средняя	Ранняя	—	—	—
Итеративный или последовательный подход		Оба	Оба			
Возможная точность		—	—			

2) *метод функциональных точек* (Functional Points), с одной стороны, прост, с другой стороны, его сложно применить в современных проектах [3].

Function Points — это единица измерения размера программных систем. Function Point Analysis (FPA) — метод определения размера программных систем (приложений) и проектов по их разработке. Размер приложения определяется с точки зрения функциональности, т. е. с точки зрения пользователя. Данный метод используется довольно редко ввиду того, что для его применения требуется полностью разработанный пользовательский интерфейс системы еще до начала разработки. С появлением Agile-подходов полная проработка интерфейсов и дизайна системы до начала разработки не делается — в результате этот метод используется редко;

3) *метод StoryPoints*. Точного перевода Story Point до сих пор не предложено. У Стива Макконелла данный метод называется «метод абстрактных рейтингов». Впервые этот метод оценки применялся в экстремальном программировании, но позже нашел широкое применение в оценке пользовательских историй в Scrum. При использовании метода абстрактных рейтингов группа оценщиков просматривает список пользовательских историй (или требований), которые они предполагают реализовать. При этом присваивает каждой истории некоторый размер, измеряемый в пунктах (*points*) или единицах сложности. При этом историям назначаются значения по одной из числовых последовательностей, представленных ниже.

#### Шкала абстрактных рейтингов

Степени 2	1, 2, 4, 8, 16
Числа Фибоначчи	1, 2, 3, 5, 8, 13, 21
Шкала М.Конна	0, 1/2, 1, 2, 3, 5, 10, 20, 40, 100

Процесс оценки проходит следующим образом: все требования записываются в виде функционального листа (Feature List); выбирается самая малая история за эталон единицы сложности — один пункт; все истории сравниваются с эталоном и друг с другом; все оценки историй складываются.

В результате получается оценка проекта в абстрактных единицах сложности, представленная ниже.

### Оценка проекта в абстрактных единицах

История	Пункт
История 1	2
История 2	1
История 3	4
...	
История 50	16
Итого	180

На этой стадии пункты не дадут никакой информации, потому что являются абстрактными единицами сложности, они не преобразуются в человеко-часы, календарное время и т. д., т. е. не приносят никакой информации для принятия решений менеджером проекта.

Почему такой формат используется?

Оценивать требования в часах (или иначе говоря, в трудозатратах) не совсем верно. Трудозатраты могут меняться исходя из производительности программиста, который будет выполнять задачу.

После проведения первичных оценок команда планирует первую итерацию, включая в нее ряд историй, т. е. планирует завершить некоторое количество абстрактных пунктов. В основу этого плана может быть заложено предположение, что один пункт равен восьми или шести человеко-часам трудозатрат, но на этом этапе это всего лишь предположение.

После завершения первой итерации у команды появляется возможность более определенно составлять дальнейшую оценку проекта (см. ниже). Команда видит, какие истории были завершены, сколько пунктов было выполнено и на основе полученной информации может строить дальнейшую оценку.

### Предварительные результаты проекта

Данные итерации № 1

Реализовано историй на 27 SP (Story Point)

Затрачено 12 человеко-недель

Затрачено 3 календарные недели

Предварительная калибровка

Объем работ = 27 SP / 12 человеко-недель = 2,25 SP/человеко-неделю

Сроки = 27 SP / 3 календарные недели = 9 SP/календарную неделю

На основе этой калибровки руководитель проекта составляет оценку оставшейся части проекта. Пример приведен ниже.

### Последующая оценка проекта

Данные итерации № 1

Предположения (из предварительной калибровки)

Объем работ = 2,25 SP/человеко-неделю

Сроки = 9 SP/календарную неделю

Размер проекта = 180 SP

Предварительная оценка для всего проекта

Объем работ = 180 SP / 2,25 SP/человеко-неделю = 80 человеко-недель

Сроки = 180 SP / 9 SP/календарную неделю = 20 календарных недель

Исходная оценка будет уточняться по данным из следующих итераций и обычно находят какое-то среднее значение производительности команды. При

этом чем короче итерации, тем быстрее у вас будут появляться данные для оценки и тем точнее будет оценка проекта;

4) *метод футболки*. Часто на ранних этапах проекта участникам проекта требуется решить, какую функциональность предстоит делать в проекте [4]. И при этом бизнес-заказчикам требуется принять решения, не обладая информацией о реальной сложности той или иной функциональности продукта. Действительно, они задаются вопросом: Как я могу понять, нужна мне эта функциональность или нет, если я не знаю, сколько она будет стоить?! На что разработчик возразит: Я не могу оценить стоимость, пока не получу более подробные требования. Ситуация заходит в тупик. Вспомним, что хорошая оценка — это та оценка, которая позволяет принимать решения для управления проектом. Для этого используется так называемый метод футболки [5].

В этом методе разработчики классифицируют все отдельные функции по отношению к другим как малые (S), средние (M), большие (L) и очень большие (XL). Параллельно бизнес-заказчики классифицируют бизнес-ценность данной функциональности, пользуясь такой же шкалой. Затем два набора оценок объединяются и сравниваются.

Именно за размеры S, M, L, XL данный метод и называется методом футболки — используются размеры футболок, представленные в табл. 2.

**Таблица 2. Классификация функций метода футболки**

Функция	Коммерческая ценность	Затраты на разработку
Функция 1	X	M
Функция 2	S	X
Функция 3	X	X
Функция 4	XL	M
Функция 5	M	XL
...		
Функция 60	M	S

Для принятия решения достаточно сравнить два столбца. Даже в таком виде таблица позволяет принимать решения: «Данная функция требует больших затрат на разработку, но обладает низкой ценностью для бизнеса — от нее лучше отказаться». Если вы исключаете работу над функцией уже на этом этапе, вы экономите не только на разработке, но и на выявлении требований, постановке задачи, управлении проектом и т. д.

Что стоит сохранить, а что выбросить? Обсуждать эту тему будет проще, если задать для размеров футболки веса и проанализировать данные по соотношению затраты/выигрыш, используя табл. 3.

**Таблица 3. Веса соответствия ценности к затратам**

Коммерческая ценность	Затраты на разработку			
	XL	X	M	S
XL	0	4	6	7
X	-4	0	2	3
M	-6	-2	0	1
S	-7	-3	-1	0

Наличие табл. 4 дает возможность добавить третий столбец в изначальную оценку и показать соотношение затрат/выгоды в числовом выражении и поставить разумный приоритет.

Таблица 4. Веса соответствия рациональности решений

Функция	Коммерческая ценность	Затраты на разработку	Решение
Функция 1	X	M	2
Функция 2	S	X	-3
Функция 3	X	X	0
Функция 4	XL	M	6
Функция 5	M	XL	-6
...			
Функция 60	M	S	1

Данную цифру не стоит воспринимать слишком серьезно и проводить черту, ниже которой функциональность не будет даже обсуждаться. Этот метод прежде всего позволяет расставить приоритеты и дать определенное *да* для функций в верхней части списка.

Этот метод обычно используется для обсуждения функциональности с заказчиком проекта.

**Сбор метрических данных. Советы, рекомендации:**

- 1) определяйте размер каждого проекта;
- 2) не усердствуйте поначалу с выбором единицы измерения — если впоследствии вам предстоит работать с реальными данными, для начала сойдут и абстрактные единицы;
- 3) стройте сложные метрики на основе простых (тех, которые легко подсчитать в любом программном продукте);
- 4) собирайте архивные данные, чтобы считать производительность труда по уже законченным проектам;
- 5) работайте над формулами вычисления сложных синтетических метрик до тех пор, пока полученные результаты не будут наиболее точно отражать отношение абстрактных единиц к указанному в архивных данных объему работ;
- 6) проведите через всю архивную базу данных линию тренда, которая будет показывать ожидаемый объем работ в виде отношения значений сложных синтетических метрик;
- 7) теперь для каждого нового проекта достаточно будет высчитать значение синтетической метрики и использовать ее при определении ожидаемого объема работ;
- 8) не забывайте об «уровне помех» на линии производительности и используйте его как индикатор при определении допустимых отклонений от общей траектории.

Проведенные научные исследования по анализу путей решений задач прогнозирования продолжительности выполнения проекта позволят сделать следующие выводы:

на основании обобщенных результатов разработан системный подход в выборе подходящего метода прогнозирования продолжительности выполнения проекта;

даны теоретические рекомендации для реализации данных методов на практике.

Проведенные в работе исследования и систематизация методов оценки продолжительности выполнения проекта позволили получить практическую и социально-экономическую значимость результатов исследования, которые могут быть использованы при расчете продолжительности программного проекта. Эти инструменты позволяют руководителям проектов лучше оценивать время и ресурсы, необходимые для завершения проекта определить задачи и

действия, которые необходимо выполнить, позволят оценить усилия, необходимые для выполнения этих задач. Это в свою очередь помогает прогнозировать общую продолжительность проекта и составлять более точные сроки проекта. Кроме того, можно использовать для государственного управления на отраслевом, региональном уровне и отдельных субъектов хозяйствования.

### Литература и электронные публикации в Интернете

1. Булова, А. Д. Управление проектами в информационных технологиях / А. Д. Булова // Весн. Беларус. дзярж. экан. ун-та. — 2020. — № 4 — С. 31–34.  
*Bulova, A. D. Upravlenie proektami v informacionnyh tehnologijah [Project Management in Information Technology] / A. D. Bulova // Vesn. Belarus. dzjarzh. jekan. un-ta. — 2020. — N 4 — P. 31–34.*
2. Schwaber, K. Agile Project Management with Scrum / K. Schwaber // Microsoft Press. — January. — 2004. — 163 p.
3. Архипенков, С. Лекции по управлению программными проектами [Электронный ресурс] / С. Архипенков // Citforum. — Режим доступа: [http://citforum.ru/SE/project/arkhipenkov\\_lectures/12.shtml](http://citforum.ru/SE/project/arkhipenkov_lectures/12.shtml). — Дата доступа: 09.02.2023.  
*Arhipenkov, S. Lekcii po upravleniju programmnyimi proektami [Lectures on software project management] [Jelektronnyj resurs] / S. Arhipenkov // Citforum. — Rezhim dostupa: [http://citforum.ru/SE/project/arkhipenkov\\_lectures/12.shtml](http://citforum.ru/SE/project/arkhipenkov_lectures/12.shtml). — Data dostupa: 09.02.2023.*
4. Методы управления проектами [Электронный ресурс]. — Режим доступа: <http://cribs.me/upravlenie-proektami/metody-upravleniya-proektami>. — Дата доступа: 09.02.2023.
5. Полковников, А. В. Управление проектами. Полный курс MBA / А. В. Полковников, М. Ф. Дубовик. — М. : Олимп-Бизнес, 2015. — 552 с.  
*Polkovnikov, A. V. Upravlenie proektami. Polnyj kurs MBA [Project management. Full MBA course] / A. V. Polkovnikov, M. F. Dubovik. — M. : Olimp-Biznes, 2015. — 552 p.*

---

**ALEXANDER BOULOVA,  
HALINA PADHORNAYA**

---

## *ESTIMATION OF IT PROJECT DURATION*

---

**Authors affiliation.** *Alexander BOULOVA* (bulavaad@belhard.com), *Belarus State Economic University (Minsk, Belarus)*; *Halina PADHORNAYA* (galpodgornaya@gmail.com), *Belarus State Economic University (Minsk, Belarus)*.

**Abstract.** The means (tools) used to calculate (predict) the duration of a project to create a software product are considered. Recommendations for their application are presented, depending on the project complexity.

**Keywords:** UseCases; User Story; Backlog; Function Point Analysis.

UDC 004.2

---

*Статья поступила  
в редакцию 24. 02. 2023 г.*