

УДК 519.8

ОБЗОР МАТЕМАТИЧЕСКИХ МЕТОДОВ В ПРОЕКТНОМ АНАЛИЗЕ ПРИ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Иванова В.Е., студент гр. ПМИ-192, IV курс

Тагунова В.Ф., студент гр. ПМИ-192, IV курс

Научный руководитель: Мешечкин В.В., кандидат физико-математических
наук, доцент

Кемеровский государственный университет
г. Кемерово

Разработка программного обеспечения – это сложный процесс, требующий тщательного анализа, планирования и управления. Проектный анализ является ключевой частью этого процесса, поскольку он позволяет разработчикам получить полное понимание того, что должно быть реализовано в проекте и каким образом это может быть достигнуто.

Математические методы играют важную роль в проектном анализе при разработке программного обеспечения. Эти методы позволяют разработчикам определить не только конкретные цели проекта, но и оптимальные пути достижения этих целей. Некоторые из самых популярных математических методов, используемых в проектном анализе, включают в себя линейное программирование, теорию вероятности, статистический анализ, теорию игр, оптимизацию, математическое моделирование и др. Линейное программирование, например, используется для определения оптимального пути достижения целей проекта при наличии ограничений. Теория вероятности позволяет оценить вероятность возникновения определенных событий, таких как возникновение рисков в проекте. Статистический анализ используется для анализа данных, полученных от пользователей, чтобы определить, каким образом они будут использовать программное обеспечение.

Одним из ключевых математических методов, используемых в проектном анализе, является статистический анализ. Статистический анализ помогает разработчикам собрать и анализировать данные о том, как пользователи будут использовать программное обеспечение. Это может включать в себя анализ того, какие функции будут наиболее полезными для пользователей, какие функции будут использоваться наиболее часто и как пользователи будут взаимодействовать с интерфейсом программного обеспечения. Методы математической статистики сводятся к систематизации, обработке и использованию статистических данных для практических выводов. Основным элементом статистического исследования является анализ и построение взаимосвязей переменных. Изучение таких взаимосвязей осложнено тем, что они не являются строгими, функциональными зависимостями. Бывает достаточно трудно выявить все основные факторы, влияющие на данную переменную (например, прибыль, риск), многие такие взаимодействия являются случай-

ными, носят неопределенный характер, и число статистических наблюдений является ограниченным. В этих условиях математическая статистика (теория обработки и анализа данных) позволяет строить экономические модели и оценивать их параметры, проверять гипотезы о свойствах экономических показателей и формах их связи, что в конечном счете служит основой для экономического анализа и прогнозирования, создавая возможность для принятия обоснованных решений. Теория вероятностей играет важную роль при статистических исследованиях вероятностно-случайных явлений. Здесь в полной мере находят применение такие основанные на теории вероятностей разделы математической статистики как статистическая проверка гипотез, статистическое оценивание распределений вероятностей и их параметров и др.

Для определения длительности проекта можно использовать статистические методы, такие как методы сетевого анализа и методы моделирования Монте-Карло.

Метод сетевого анализа позволяет определить зависимости между задачами проекта, определить критические пути и оценить продолжительность проекта. В данном методе первым шагом идет определение задач и их последовательности, для каждой задачи определяется ее продолжительность и зависимости от других задач. Зависимости могут быть двух типов: начальные (предшествующие задачи) и конечные (следующие задачи). После определения задач на их основе следует построение сетевой диаграммы, которая представляет собой представление задач и их зависимостей. Задачи представлены в виде узлов, а зависимости – в виде стрелок, указывающих на направление выполнения задачи. Начальные задачи обычно располагаются в левой части диаграммы, а конечные – в правой. Построив сетевую диаграмму, по ней ищут длину критического пути, который является последовательностью задач, определяющих продолжительность проекта. Критический путь проходит через задачи, которые не могут быть начаты до завершения предыдущих задач. Определение критических путей позволяет оценить минимальную продолжительность проекта и выявить наиболее критические задачи, которые могут задержать завершение проекта. Критический путь определяется резервы времени, которые являются запасом времени между завершением задачи и началом следующей задачи. Резерв времени может использоваться для ускорения проекта или компенсации возможных задержек. Для поиска временных резервов рассчитываются ранние и поздние начала и окончания работ проекта. Ранние начала рассчитываются на основе последовательности задач и времени, которое они занимают. Кратко говоря, раннее начало задачи соответствует длительности предыдущих работ:

$$T_i^{\text{PH}} = T_{i-1}^{\text{PH}} + D_{ij-1}, \quad (1)$$

где T_i^{PH} – время раннего начала работы, T_{i-1}^{PH} – раннее начало предыдущей работы, D_{ij-1} – длительность предыдущей работы;

Раннее окончание работы соответствует сумме длительности текущей работы и ее раннему началу:

$$T_i^{PO} = D_{ij} + T_i^{PH}, \quad (2)$$

где T_i^{PO} – время раннего окончания работы, D_{ij} – длительность текущей работы, T_i^{PH} – время раннего начала работы;

При расчете поздних начал и окончаний начинают именно с окончаний. Расчет позднего окончания свидетельствует о начале расчета обратного прохода, смысл которого заключается в том, чтобы вести расчеты начиная с первой работы, а не с последней. Времени позднего окончания последней работы соответствует его раннее окончание, а последующих работ – минимальная разница из значений, идущих после текущей работы и их длительностью:

$$T_i^{PO} = \min_i \{T_i^{PO} - D_{ij}\}, \quad (3)$$

где T_i^{PO} – время позднего окончания работы, D_{ij} – длительности операций;

Позднее начало соответствует разнице между поздним окончанием и длительностью работы:

$$T_{ij}^{PH} = T_j^{PO} - D_{ij}, \quad (4)$$

где T_{ij}^{PH} – время позднего начала работы, T_j^{PO} – время позднего окончания работы, D_{ij} – длительности операций.

Определив ранние и поздние начала и окончания, рассчитываются резервы. Существуют полный и свободный резервы и когда они при одной работе равны нулю, то данная работа является входящей в критический путь. Для определения полного резерва ищется разница между поздним окончанием, ранним началом и длительностью работы:

$$PR_{ij} = T_j^{PO} - T_i^{PH} - D_{ij}, \quad (5)$$

где PR_{ij} – полный резерв времени, T_j^{PO} – время позднего окончания работы, T_i^{PH} – время раннего начала работы, D_{ij} – длительности операций.

Свободный резерв определяется разницей между максимальным из значений ранних окончаний, ведущих к одному узлу, и ранним окончанием текущей работой:

$$SP_{ij} = \max_j (T_j^{PO}) - T_i^{PO}, \quad (6)$$

где SP_{ij} – свободный резерв времени, T_i^{PO} – раннее окончание.

На основе продолжительности задач, критических путей и резервов времени определяются даты начала и завершения каждой задачи проекта. Это позволяет оценить общее время выполнения проекта и управлять его продолжительностью.

Метод Монте-Карло – это стохастический метод моделирования, который позволяет оценить вероятность возможных исходов событий на основе случайных выборок. В проектном анализе при разработке программного обеспечения метод Монте-Карло используется для прогнозирования возможных результатов проекта, оценки рисков и принятия решений на основе вероятностных моделей.

Вот некоторые методы Монте-Карло, которые используются в проектном анализе при разработке программного обеспечения:

1. Оценка длительности проекта. Данный метод используется для прогнозирования длительности проекта. Проект разбивается на несколько этапов и для каждого этапа определяются возможные варианты длительности. Далее строятся вероятностные модели, на основе которых определяется вероятность возникновения каждого варианта длительности. Путем случайной выборки из этих моделей определяется распределение вероятностей длительности проекта. Формула для вычисления вероятности возникновения каждого варианта длительности при оценке длительности проекта методом Монте-Карло может быть представлена следующим образом:

$$P(X = x) = \frac{n(x)}{N}, \quad (7)$$

где $P(X = x)$ – вероятность возникновения определенного значения длительности проекта (x), $n(x)$ – количество случаев, когда длительность проекта оценивалась как x в процессе моделирования методом Монте-Карло, N – общее количество итераций моделирования.

Для применения данной формулы необходимо провести достаточное количество итераций моделирования методом Монте-Карло, чтобы убедиться в получении достаточно точных оценок вероятности для каждого варианта длительности проекта.

2. Оценка бюджета проекта. Данный метод используется для прогнозирования стоимости проекта. Проект разбивается на несколько этапов и для каждого этапа определяются возможные затраты. Далее строятся вероятностные модели, на основе которых определяется вероятность возникновения каждого варианта затрат. Путем случайной выборки из этих моделей определяется распределение вероятностей стоимости проекта. Формула для вычисления вероятности возникновения каждого варианта затрат, будет следующей:

$$P = \frac{N_i}{N}, \quad (8)$$

где P – вероятность возникновения варианта затрат, N_i – количество случаев, когда случайное число попало в интервал для данного варианта затрат, N – общее количество испытаний (например, 10 000).

3. Оценка рисков. Проект разбивается на несколько этапов, для каждого этапа определяются возможные риски и вероятность их возникновения. Далее строятся вероятностные модели, на основе которых определяется вероятность возникновения каждого варианта риска. Путем случайной выборки из этих моделей определяется распределение вероятностей рисков проекта. Для вычисления рисков методом Монте-Карло формула будет иметь следующий вид:

$$P = \frac{m}{N} \cdot 100\%, \quad (9)$$

где P – вероятность возникновения риска (в процентах), m – количество случаев, когда риск проявился в ходе моделирования, N – общее количество испытаний.

Для применения метода Монте-Карло необходимо определить вероятностные распределения для каждого из факторов, влияющих на вероятность

возникновения риска. Затем генерируются случайные значения для каждого фактора в соответствии с его распределением. На основе полученных значений производится моделирование возможных исходов и оценка вероятности возникновения риска. Чем больше число случаев моделирования, тем более точной будет полученная оценка вероятности.

4. Определение оптимальной стратегии разработки. Проект разбивается на несколько этапов, для каждого этапа определяются возможные варианты стратегий разработки. Далее строятся вероятностные модели, на основе которых определяется вероятность достижения каждого варианта целевых показателей (например, сроков разработки, качества программного обеспечения и т.д.). Путем случайной выборки из этих моделей определяется оптимальная стратегия разработки программного обеспечения. Для вычисления вероятности достижения каждого варианта целевых показателей при определении оптимальной стратегии разработки методом Монте-Карло можно использовать следующую формулу 9, где m – количество наборов входных параметров, для которых целевые показатели были достигнуты.

5. Оценка качества программного обеспечения. Данный метод используется для оценки качества программного обеспечения на основе тестирования. На каждом этапе разработки определяются возможные дефекты, которые могут возникнуть, и вероятность их возникновения. Далее строятся вероятностные модели, на основе которых определяется вероятность обнаружения каждого дефекта и вероятность его исправления. Путем случайной выборки из этих моделей определяется распределение вероятности качества программного обеспечения.

6. Определение оптимального размера команды разработки. Проект разбивается на несколько этапов, и для каждого этапа определяется требуемый объем работ. Далее строятся вероятностные модели, на основе которых определяется вероятность успешного завершения проекта для различных размеров команды разработки. Путем случайной выборки из этих моделей определяется оптимальный размер команды разработки.

Другим важным математическим методом, используемым в проектном анализе, является алгоритмический анализ. Алгоритмический анализ помогает разработчикам определить наилучшие методы реализации программного обеспечения, используя определенные алгоритмы и структуры данных. Это может включать в себя определение наилучшего алгоритма для выполнения определенной операции или определения наилучшей структуры данных для хранения и обработки определенных типов информации.

Процесс алгоритмического анализа можно разделить на следующие шаги. Первым шагом является определение задачи, которую необходимо решить, и выбора алгоритма, который будет использоваться для ее решения. Следующим шагом является оценка времени выполнения алгоритма. Это можно сделать, например, путем подсчета количества операций, которые необходимо выполнить в процессе работы алгоритма. Важно также оценить, какие ресурсы (например, память) будут потребляться в процессе выполнения

алгоритма. На основе оценки времени выполнения и потребления ресурсов можно провести анализ сложности алгоритма. Сложность алгоритма определяет, насколько быстро он будет работать при увеличении объема входных данных. В зависимости от результатов анализа сложности, необходимо проанализировать другие алгоритмы, которые могут использоваться для решения данной задачи. Оценить их эффективность и выбрать наиболее подходящий. После выбора наиболее подходящего алгоритма необходимо провести тестирование его работоспособности и производительности. В конце процесса необходимо проанализировать результаты тестирования и сравнить их с ожидаемыми. Если результаты удовлетворительные, то алгоритм может быть использован для решения задачи.

Также математические методы могут использоваться для определения рисков и управления ими в проектном анализе. Например, методы теории вероятности могут быть использованы для определения вероятности возникновения определенного риска и его влияния на проект, а методы оптимизации могут быть использованы для определения наилучшего способа управления рисками.

Ниже представлены некоторые методы теории вероятностей:

1. Распределение вероятностей – метод, позволяющий оценить вероятность того, что случайное событие произойдет в течение заданного временного интервала. Распределение вероятностей может быть использовано не только для определения вероятности возникновения какого-либо риска, но и для определения вероятности успешной реализации проекта в течение определенного периода времени.

2. Моделирование событий – метод, который используется для определения вероятности того, что определенное событие произойдет в определенной ситуации. Моделирование событий может использоваться для прогнозирования рисков, связанных с разработкой программного обеспечения.

3. Марковские процессы – метод, который используется для моделирования последовательности событий, которые могут произойти в будущем. Марковские процессы могут использоваться для прогнозирования рисков, связанных с разработкой программного обеспечения и определения вероятности успешной реализации проекта.

4. Байесовский подход – метод, который используется для определения вероятности того, что определенное событие произойдет, основываясь на предварительных знаниях. Байесовский подход может использоваться для прогнозирования рисков, связанных с разработкой программного обеспечения и определения вероятности успешной реализации проекта.

Оптимизация проекта в разработке программного обеспечения – это процесс повышения эффективности и производительности разработки, путем применения различных методов и инструментов. Методы оптимизации направлены на ускорение процесса разработки, улучшение качества продукта и повышение эффективности работы команды разработчиков. Ниже пред-

ставлены более подробно несколько основных методов оптимизации при разработке программного обеспечения:

1. Метод Agile – это итеративный подход к разработке программного обеспечения, который основан на гибкости и быстродействии. Он позволяет команде разработки быстро адаптироваться к изменениям и требованиям заказчика, что ускоряет процесс разработки и повышает качество программного продукта. Agile основывается на принципах, таких как самоорганизация, коммуникация и сотрудничество.

2. Метод DevOps – это подход к разработке программного обеспечения, который объединяет разработку и эксплуатацию в единый процесс. DevOps позволяет сократить время между разработкой и выпуском продукта на рынок, а также обеспечивает более высокую стабильность и надежность при работе с программным обеспечением.

3. Метод Continuous Integration – это процесс объединения всех изменений в коде в единое целое, путем автоматической сборки и тестирования. Это позволяет быстрее обнаруживать и исправлять ошибки, сокращает время на внедрение новых функций и улучшений в продукт, а также повышает качество программного обеспечения.

4. Метод Continuous Delivery – это метод, который автоматически доставляет изменения в коде в производственную среду после прохождения всех тестов и контроля качества. Этот процесс позволяет ускорить время доставки продукта на рынок и улучшает качество программного обеспечения.

5. Метод автоматического тестирования – это метод, при котором тесты проводятся автоматически, что позволяет сократить время на их проведение и повысить качество программного обеспечения.

В целом, использование математических методов в проектном анализе при разработке программного обеспечения является необходимым условием для создания продукта высокого качества, который будет соответствовать требованиям пользователей и будет выполнять свои функции в соответствии с поставленными целями. Эти методы позволяют разработчикам получить полное понимание требований проекта, определить оптимальные пути достижения целей проекта, а также оценить риски и способы их управления.

Список литературы:

1. Проектный анализ. Оценка реализуемости проекта [Электронный ресурс]. URL: <https://studfile.net/preview/9591088/page:14> (Дата обращения: 31.03.2023).

2. Проектный анализ: основные принципы, этапы и виды [Электронный ресурс]. URL: https://www.cfin.ru/finanalysis/invest/project_analysis.shtml (Дата обращения 30.03.2023).

3. Шкурко, В.Е. Управление рисками проекта: учебное пособие для вузов / В.Е. Шкурко – М.: Юрайт, 2022. – 182 с.

4. Ньюэлл, М.В. Управление проектами. Руководство по подготовке к сдаче сертификационного экзамена / Майкл В. Ньюэлл. – М.: Кудиц, 2006. – 416 с.