ЧАСТНАЯ ЗАДАЧА ОПТИМИЗАЦИИ СРОКОВ ПРИ УПРАВЛЕНИИ ПРОЕКТАМИ

Маркаров Д.А.

(*Институт проблем управления РАН, Москва*)

Введение

В соответствии с определением, приведенным в [2], «*проект* – ограниченное во времени целенаправленное изменение отдельной системы с установленными требованиями к качеству результатов, возможными рамками расхода средств и ресурсов и специфической организацией».

На сегодняшний день теория *управления проектами* (УП) является бурно развивающимся разделом теории управления социально-экономическими системами.

Ниже в статье содержится описание частной модели управления проектом по разработке программного обеспечения и решение в ее рамках задачи о минимизации сроков.

Постановка задачи

Проектом является разработка программного продукта, обладающего функциональностью, необходимой заказчику. Таким образом, выделяются две фазы проекта: сбор информации о требованиях заказчика аналитиком и собственно разработка программного продукта программистом. Человеческие ресурсы заданы, бюджет проекта не учитываем. Полагается, что передача проекта в разработку может быть осуществлена до полного сбора информации о требованиях, но затраты времени на разработку тогда возрастают.

Задача минимизации сроков выполнения проекта – это задача поиска момента времени инициация разработки, при котором сроки выполнения будут минимальны.

Для формирования модели представим, что проект есть совокупность N требований, которая должна быть полностью выявлена и выполнена. Для выявления аналитику требуется время $T_Y = t_Y N$,

где t_Y – время, требующееся на выявление одного требования. Считаем, что интенсивность сбора требований одинакова во время всего выполнения задачи. Аналогично программисту для реализации всего проекта требуется время $T_W = t_W N$.

Таким образом, при «нормальном» выполнении проекта, когда все требования сначала будут собраны, а затем реализованы, его длительность составит $T_Y + T_W$.

Рассмотрим вариант, в котором реализация проекта начинается до сбора всех требований. Пусть аналитик, собрав K требований из N, передает их на выполнение программисту, а по окончании сбора передает оставшиеся $(N - K)$ требований. Будем считать, что в этом случае на выполнение каждого из первых K требований программисту понадобится то же время t_W, что и при обычной работе, а на реализацию каждого из оставшихся $(N -)$ без потери качества необходимо $t'_W = t_W (1 + l)\dfrac{N-K}{N}$ времени, где l – коэффициент «усложнения». Задача заключается в нахождении такого значения K, чтобы общая длительность проекта сократилась по сравнению с «нормальным» выполнением.

Решение задачи

Итак, из постановки задачи видно, что при досрочной разработке проекта, реализация начинается на $(N - K)\,t_Y$ единиц времени раньше, а программист затратит время Kt_W на реализацию первых K требований. Экономия времени $\Delta T_- = \min((N-K)t_Y; Kt_W)$. При этом дополнительно будет затрачено время $\Delta T_+ = t_W\, l\, \dfrac{N-K}{N}(N-K)$. Для того, чтобы K удовлетворяло условиям задачи, необходимо и достаточно выполнения следующего неравенства: $\Delta T_- > \Delta T_+$.

Таким образом, решение задачи сводится к решению двух систем неравенств. Рассмотрим первую, учитывая, что в этом случае $\Delta T_- = (N-K)t_Y$:

(1) $(N-K)t_Y > t_W l \dfrac{N-K}{N}(N-K)$,

(2) $Kt_W > (N-K)t_Y$.

После преобразования получаем:

(1*) $\dfrac{t_Y}{t_W} > l \dfrac{N-K}{N}$,

(2*) $\dfrac{t_Y}{t_W} < \dfrac{K}{N-K}$.

Пусть $T = \dfrac{t_Y}{t_W}$, при дальнейшем упрощении получаем:

(1**) $K > N(1-\dfrac{T}{l})$,

(2**) $K > N\dfrac{T}{1+T}$.

Обращаем внимание, что $\Delta T_{-} = (N-K)t_Y$ будет максимально при минимальном *K*, *K* не превышает *N*. Из дальнейшего рассмотрения получаем для этого случая решение:

$$T \in (0;\frac{\sqrt{1+4l}-1}{2}) \Rightarrow N(1-\frac{T}{l}) < K < N,$$

$$T \in (\frac{\sqrt{1+4l}-1}{2};\infty) \Rightarrow N(\frac{T}{1+T}) < K < N.$$

Рассмотрим вторую систему неравенств, учитывая, что в этом случае $\Delta T_{-} = Kt_W$:

(3) $Kt_W > t_W l \dfrac{N-K}{N}(N-K)$,

(4) $(N-K)t_Y > Kt_W$.

После первого преобразования получаем:

(3*) $\dfrac{K}{N-K} > l \dfrac{N-K}{K}$,

(4*) $\dfrac{t_Y}{t_W} > \dfrac{K}{N-K}$.

Пусть $T = \dfrac{t_Y}{t_W}$, при дальнейшем преобразовании получаем:

(3**) $K^2 + \dfrac{2l-1}{l}KN + N^2 < 0$,

(4**) $K < N\dfrac{T}{1+T}$.

Получаем решение для этого случая:

$$K \in ((\frac{1-2l}{2} - \sqrt{(\frac{2l-1}{l})^2 - 4N^2}; \frac{1-2l}{2} + \sqrt{(\frac{2l-1}{l})^2 - 4N^2}) \cap$$

$$\cap (0; N\frac{T}{1+T}))$$

Общее решение будет состоять из объединения решения двух систем.

Хотелось бы отдельно отметить, что если у второй системы неравенств существование решения ограничено многими условиями, то у первой оно существует при большом количестве возможных значений параметров.

Обобщение модели и задача о существовании решения

Мы получили решение для конкретной функции, теперь рассмотрим общую модель. В ней мы поставим задачу нахождения условий, при которых существует решение задачи минимизации сроков проекта, отличное от «нормального» выполнения проекта.

Для формирования общей модели представим, что проект есть совокупность *N* требований, которая должна быть полностью выявлена и выполнена. Пусть эффективность выявления требований аналитиком описывается функцией *y*(*t*), причем выявление *N* требований составляет объем работы *Y*. Считаем, что эффективность сбора зависит только от времени, затраченного на работу. Аналогично эффективность реализации проекта программистом описывается функцией $w(N, K, t)$, где N – общее количество требований, K – количество переданных на реализацию требований.

Будем считать, что при «нормальном» выполнении проекта, когда все требования сначала будут собраны, а затем реализованы, программисту потребуется T_W времени, а общая длительность составит $T_Y + T_W$.

Формально выпишем ограничения, описывающие задачу, считая $t = 0$ отдельно для аналитика и программиста в начале выполнения их задач:

$$\int_0^{T_Y} y(t)dt = Y,$$

$$\int_0^{T_W} w(N, N, t)dt = W.$$

Рассмотрим вариант, в котором реализация проекта начинается до сбора всех требований. Пусть аналитик, собрав K требований из N, что составило объем работ X и заняло T_K времени, передает их на выполнение программисту, а по окончании сбора передает оставшиеся $(N - K)$ требований. Получается следующее:

$$\int_0^{T_K} y(t)dt = X,$$

$$\int_0^{T_Y - T_K} w(N, K, t)dt + \int_{T_Y - T_K}^{T'_W} w(N, N, t)dt = W.$$

Определим условия, при которых существует такое K, чтобы общая длительность проекта сократилась по сравнению с «нормальным» выполнением, то есть $T'_W < T_W$.

Рассмотрим граничную ситуацию: $T'_W = T_W + T_Y - T_K$. Тогда мы имеем:

$$\int_0^{T_Y - T_K} w(N, K, t)dt + \int_{T_Y - T_K}^{T'_W} w(N, N, t)dt = W =$$

$$= \int_0^{T_Y - T_K} w(N, K, t)dt + \int_{T_Y - T_K}^{T_W} w(N, N, t)dt = W$$

Таким образом, условие существования решения задачи можно сформулировать следующим способом: если за время T_W сделано больше работы, чем W, то на работу W понадобится меньше времени. Формально это записывается следующим образом:

$$(5) \quad \int_0^{T_Y - T_K} w(N, K, t)dt + \int_{T_Y - T_K}^{T_W} w(N, N, t)dt > W$$

Отсюда следует:

$$\int_0^{T_Y - T_K} w(N, K, t)dt + \int_{T_Y - T_K}^{T_W} w(N, N, t)dt > \int_0^{T_W} w(N, N, t)dt$$

или

$$\int_0^{T_Y - T_K} w(N, K, t)dt > \int_0^{T_Y - T_K} w(N, N, t)dt$$

Обозначим первообразную по t $F_t(t)$ и ограничим вид функций следующими условиями:

$$w(N, K, 0) = w(N, N, 0) = w(N, 0, 0) = 0, \; F_t(w(N, K, t))(0) = 0,$$

$$y(0) = 0, \; F_t(y(t))(0) = 0.$$

Тогда условие существования решения сведется к следующему:

$$F_t(y(t))(T_K) = X, \; F_t(y(t))(T_Y) = Y,$$

$$F_t(w(N, K, t))(T_Y - T_K) > F_t(w(N, N, t))(T_Y - T_K).$$

Само решение будет получено следующим образом: $F_t(y(t))(T_K) = X$, что позволяет нам искать не собственно K, а время T_K. Далее:

$$F_t(w(N, K, t))(T_Y - T_K) - F_t(w(N, N, t))(T_Y - T_K) +$$

$$+ F_t(w(N, N, t))(T'_W) = W, \; T'_W - T_K \to \min$$

Таким образом, еще до решения задачи можно определить, существует ли у нее решение. Само решение ищется в каждом конкретном случае исходя из специфики вида функций.

Содержательная интерпретация и недостатки модели

Решение задачи предполагает предварительную оценку сложности проекта. Выше в статье предлагается математическая модель, в которой оценка уже задана, причем она достоверна (главный недостаток модели). Для заданных функций задача точно решается и приводятся содержательные интерпретации полученных результатов.

Содержательно, рассматриваемая при данных ограничениях задача представляет собой выбор оптимального управления проектом, выполняемого внутренним ИТ-отделом компании, задачей которого является максимально быстрое и соответствующее проблемам заказчика (другого отдела компании) решение проблемы по автоматизации деятельности. Предлагаемая модель позволяет оценить различные варианты ведения проекта, нагрузку на людей, выполняющих проект и время, затраченное на выполнение.

Дальнейшее развитие модели – оценка оперативного управления проектом в зависимости от изменяющейся оценки сложности, рассмотрение общего вида функций затрат времени, переход к непрерывному случаю.

Литература

1. Коновальчук Е.В., Новиков Д.А. *Модели и методы оперативного управления проектами. М.: ИПУ РАН, 2004.*
2. Воропаев В.И. *Управление проектами в России. М.: Аланс, 1995.*