

COMPUTER-AIDED ENGINEERING

СИСТЕМЫ АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ

УДК 355.01:004.056

А.А. Перцев, А.Н. Подобрый, Ю.А. Радионова

МОДЕЛЬ РАСЧЕТА ДЛИТЕЛЬНОСТИ ИЗГОТОВЛЕНИЯ АППАРАТНО-ПРОГРАММНЫХ ИЗДЕЛИЙ, РАЗРАБАТЫВАЕМЫХ ПРОЕКТНОЙ ОРГАНИЗАЦИЕЙ

Перцев Андрей Алексеевич, кандидат технических наук, окончил механико-математический факультет Ульяновского государственного университета. Начальник отдела ФНПЦ АО «НПО «Марс». Имеет статьи в области внедрения автоматизированной системы управления предприятием. [e-mail: mars@mv.ru].

Подобрый Александр Николаевич, кандидат технических наук, окончил механико-математический факультет УлГУ. Заместитель начальника отдела ФНПЦ АО «НПО «Марс». Имеет статьи в области внедрения автоматизированной системы управления предприятием. [e-mail: mars@mv.ru].

Радионова Юлия Александровна, кандидат технических наук, окончила механико-математический факультет УлГУ. Ведущий инженер-программист ФНПЦ АО «НПО «Марс». Имеет публикации в сфере автоматизированных систем документооборота, интеллектуальной организации хранилищ технической документации, статистической оценки поставщиков. Сфера научных интересов: электронный документооборот, архивохранилища, статистический анализ данных, системы поддержки принятия решений. [e-mail: julia-owl@mail.ru].

Аннотация

Для проектной организации, выпускающей сложные аппаратно-программные комплексы (АПК), актуальным является предварительная оценка длительности разработки АПК, а также определение основных факторов, влияющих на успешность разработки. Для работ, выполняемых по серийной или серийной с мелкими изменениями схемам, оценка может быть дана на основании существующего опыта. Однако при проведении предварительного оценивания могут быть не учтены какие-либо факторы, которые не оказывали большого влияния в предыдущие периоды.

В статье представлен подход к расчету длительности изготовления АПК на основе аналогов, статистических оценок и выделения факторов, влияющих на ход изготовления. В то же время необходимо отметить, что данный подход позволяет дать только предварительную оценку и его использование для повседневной работы требует дополнительных исследований.

Ключевые слова: длительность формирования плана производства, нейронная сеть, мелкосерийное производство, проектное производство, машиностроение, мощность производства, статистика выполнения операций.

doi: 10.35752/1991-2927-2021-3-65-67-76

THE CALCULATION MODEL OF HARDWARE AND SOFTWARE SYSTEM PRODUCTION PERIOD DESIGNED BY DEVELOPING COMPANIES

Andrei Alekseevich Pertsev, Candidate of Sciences in Engineering; graduated from the Faculty of Mechanics and Mathematics of Ulyanovsk State University; Head of a department of FRPC JSC 'RPA 'Mars'; an author of articles in the field of the automated enterprise management system implementation. e-mail: mars@mv.ru.

Aleksandr Nikolaevich Podobrii, Candidate of Sciences in Engineering; graduated from the Faculty of Mechanics and Mathematics of Ulyanovsk State University; Deputy Chief of a department of FRPC JSC 'RPA 'Mars'; an author of articles in the field of the automated enterprise management system implementation. e-mail: mars@mv.ru.

Iuliia Aleksandrovna Radionova, Candidate of Sciences in Engineering; graduated from the Faculty of Mechanics and Mathematics of Ulyanovsk State University; Lead Programming Engineer at FRPC JSC 'RPA 'Mars'; an author of articles in the field of automated workflow systems, intelligent technical documentation storage bases and systems for statistical analysis of supplier appraisal; research interests are in the field of electronic document management, archival depositories, statistical data analysis, decision support systems. e-mail: julia-owl@mail.ru.

Abstract

It is relevant for a developer of hardware and software systems to assess the duration of development in advance and determine the main factors influencing on its success. Bulk production or bulk production with little changes can be assessed based on the experience. However, when making preliminary assessment, some factors that had not a great effect before may not be considered.

The article describes an approach to calculate the duration of hardware and software system development based on analogues, statistics and factors effecting the production. At the same time, it should be noted this method provides only a preliminary assessment and its usage requires additional s.

Keywords: duration of production plan development, neural net, small-scale manufacturing, design production, machinery, production capacity, operation flow statistics.

ВВЕДЕНИЕ

Аппаратно-программный комплекс (АПК) – продукт, в состав которого входят технические средства (ТС) и программное обеспечение (ПО), совместно применяемые для решения задач определенного типа. Общая схема состава работ по изготовлению АПК представлена на рисунке 1.

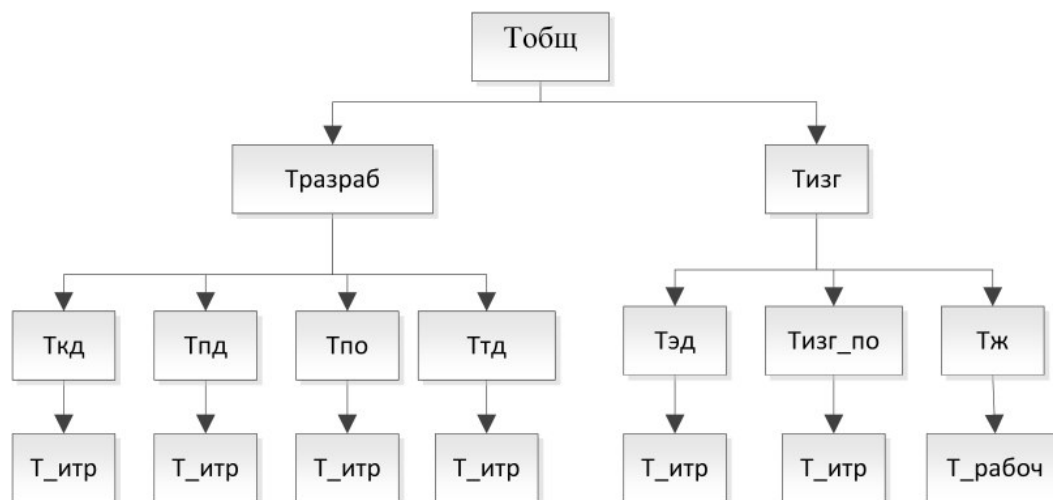
Основной задачей проектной организации, выпускающей АПК, является своевременное изготовление продукции надлежащего качества. Для обеспечения

процесса производства зачастую используются различные наборы планов: от стратегического до детализирующего плана участка производственного подразделения. Если рассматривать сложные АПК, то их производство распадается на две основные части: разработку ТС и ПО, которые могут быть выполнены параллельно. Таким образом, можно рассматривать три основных этапа: изготовление ТС АПК, изготовление ПО и комплексование ПО на ТС, после чего уже готовое изделие может быть сдано заказчику и передано в эксплуатацию.



Рис. 1. Общая схема состава работ АПК

Исходя из состава работ длительность процесса разработки и изготовления АПК с учетом разделения по специалистам можно представить следующим образом:



Здесь Тобщ – общий срок изготовления изделия;
 Тразраб – время, требующееся на разработку проекта;
 Тизг – время, требующееся на изготовление;
 Ткд – время разработки конструкторской документации (КД);
 Тпд – время разработки программной документации (ПД);
 Тпо – время разработки ПО;
 Ттд – время разработки технологической документации (ТД);
 Тэд – время изготовления эксплуатационной документации (ЭД);
 Тизг_по – время изготовления ПО;
 Тж – время изготовления аппаратной части;
 Т_итр – временная нагрузка инженерно-технических работников;
 Т_работч – временная нагрузка рабочих специалистов.

Для решения основной задачи проектной организации можно сформулировать задачу определения срока изготовления ТС с использованием данных различных автоматизированных систем (АС) предприятия, объединенных в общую систему, обладающую свойствами онтологии.

В общем случае задачу определения срока можно разбить на две подзадачи:

- 1) полное изготовление нового ТС с полной разработкой документации;
- 2) изготовление ТС, для которого часть документации и материальных составляющих может быть в наличии, а часть – должна быть разработана (откорректирована) и изготовлена.

При разработке новых (доработке имеющихся) узлов ТС на любом уровне вложенности система должна предоставлять информацию об изменении сроков изготовления. Также при разработке новых ТС координатор процесса должен получить информацию по срокам

изготовления ТС, подобных новому по свойствам или функциональности.

Для каждого изделия (составной части, узла) существует определенный набор операций по его изготовлению. Для каждой операции, в свою очередь, существует значение норматива трудоемкости и перечень должностей (специалистов), которые могут ее выполнить. Корректировки свойств объектов данной части онтологии также дают сигнал для корректировки трудоемкости изготовления изделия либо сроков выполнения. Например, изменение в штате сотрудников (увольнения, отпуска) может повлиять на сроки изготовления изделия в сторону увеличения.

Общая схема структуры АС предприятия, используемых в процессе изготовления ТС, представлена на рисунке 2 (Общая схема данных производственных подсистем по формированию план-графиков изготовления ТС).

1 СОСТАВ ДОКУМЕНТАЦИИ НА ИЗДЕЛИЕ В АРХИВЕ ТЕХНИЧЕСКОЙ ДОКУМЕНТАЦИИ ПРОЕКТНОЙ ОРГАНИЗАЦИИ

В процессе жизненного цикла (ЖЦ) любого изделия, выпускаемого проектной организацией, накапливаются большие массивы информации как структурированной, так и хаотической, и необходимо наличие механизмов ее накопления, хранения и обработки. Одним из подобных механизмов может служить электронный архив, в составе которого большую часть занимают КД, ПД и ТД, используемые при изготовлении изделий [1]. Обобщенная структура формирования архивной документации на изделие представлена на рисунке 3.

Спецификация изделия фактически представляет собой описание структуры изделия – документация, необходимая для изготовления, эксплуатации и ремонта изделия: количество составных частей (деталей и сборочных единиц); материалы; комплектующие и так далее.

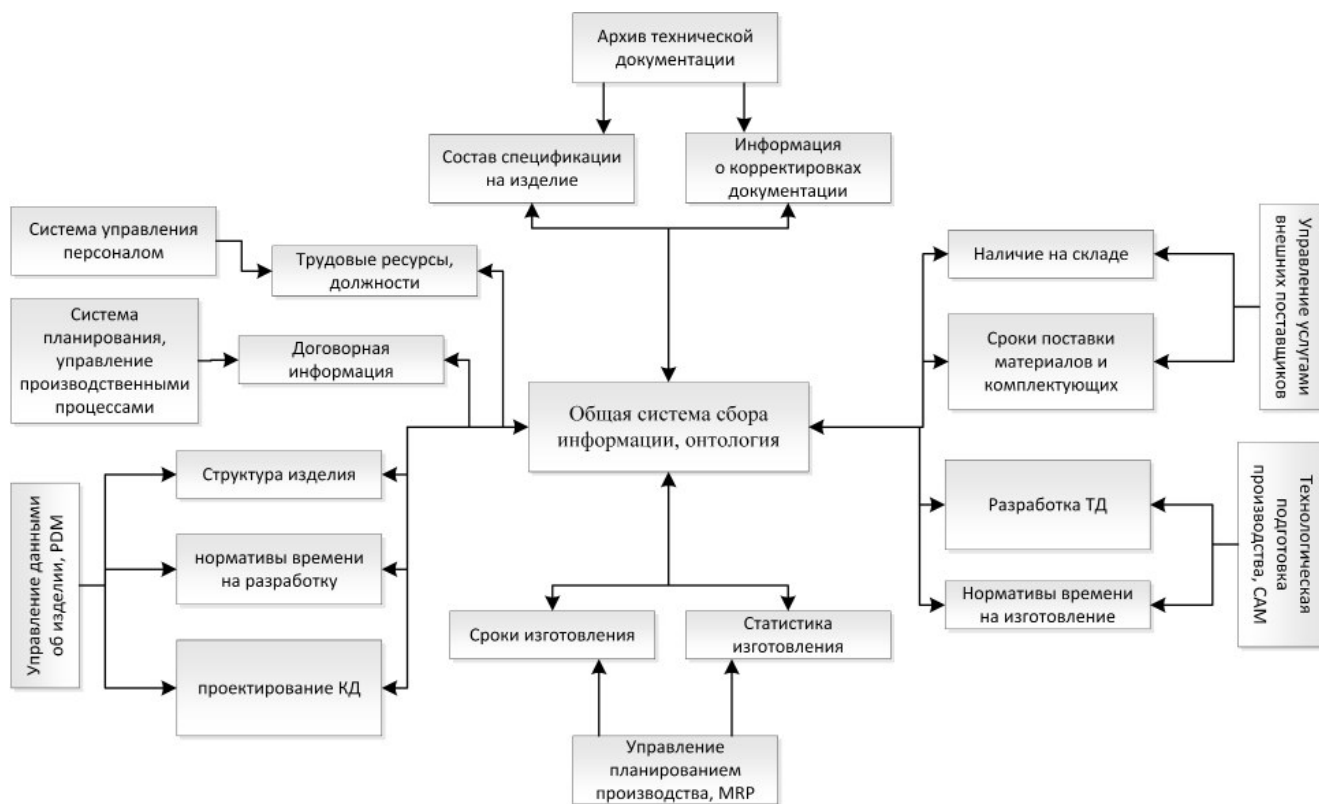


Рис. 2. Структура АС предприятия



Рис. 3. Структура формирования архивной документации на изделии

В свою очередь, спецификация каждой составной части изделия может представлять собой самостоятельное изделие, программное средство или ТС и состоять из такого же (или меньшего) количества составных частей.

Необходимость наличия документации на изделие в едином архиве предприятия обусловлена следующими требованиями:

- 1) осуществление контроля полноты разрабатываемого комплекта, то есть наличие всех документов, необходимых для изготовления изделия и записанных в спецификации [2];
- 2) проведение нормоконтроля документации;

3) исключение потери документа либо какой-либо из версий документа, обеспечение наличия всех исходных текстов ПО, инструкций и описаний, в том числе для сборки и компиляции ПО;

4) при наличии электронного архива – возможность одновременного использования одного документа разными пользователями, а также своевременные оповещения о корректировках документов.

Структура базы данных архива позволяет устанавливать связи между спецификацией и входящими в нее документами, что, в свою очередь, дает возможность сделать автоматический подбор полного комплекта документации по изделию. В рамках производственного процесса изготовления изделия включение подобной структуры в общую систему (онтологию) позволит оперативно отслеживать корректировку документации на любом уровне вложенности, что, в свою очередь, дает возможность оперативно корректировать сроки и порядок изготовления.

2 МЕТОД ОЦЕНКИ ТРУДОЕМКОСТИ РАЗРАБОТКИ ПО

Планирование разработки ПО – процесс ЖЦ, позволяющий оценить трудоемкость и длительность разработки ПО. Точность планирования важна для управления проектом разработки ПО и определения трудоемкости и длительности всего проекта в целом. Для определения трудоемкости могут использоваться различные методы: регрессионный анализ, статистические модели, генетические алгоритмы, нечеткая логика

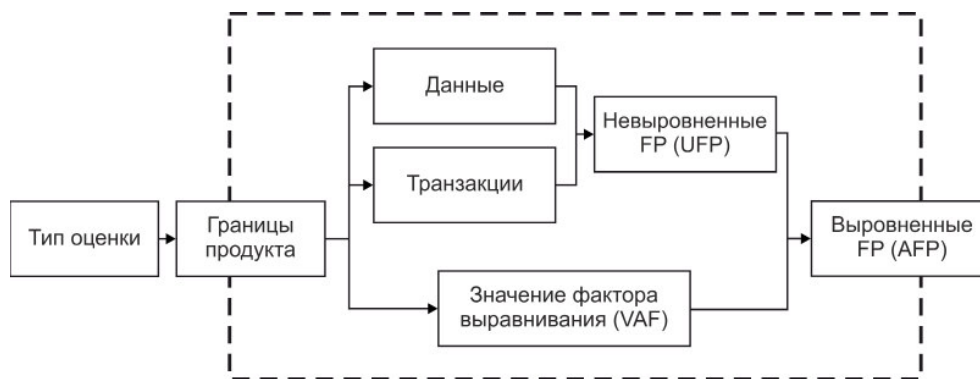


Рис. 4. Последовательность действий для оценки по методу анализа функциональных точек

и нейронные сети. Однако проблема точности определения трудоемкости до настоящего времени не решена.

Для решения оценки длительности разработки ПО широкое распространение получили несколько методов. Каждый из методов направлен на преодоление неопределенности длительности разработки – выявление факторов, влияющих на процесс разработки.

Анализ функциональных точек – стандартный метод измерения размера программного продукта с точки зрения пользователей системы. Метод разработан Аланом Альбрехтом (Alan Albrecht) в середине 70-х. Метод был впервые опубликован в 1979 году. Метод предназначен для оценки на основе логической модели объема программного продукта: количество функционала, востребованного заказчиком и разрабатываемого исполнителем.

Основным достоинством метода является то, что измерения не зависят от технологической платформы, на которой будет разрабатываться продукт, и он обеспечивает единообразный подход к оценке всех проектов организации. Последовательность действий приведена на рисунке 4. Подробное описание метода можно найти в первоисточнике [3], либо в изложении отечественных авторов [4].

Основным недостатком является то, что метод ничего не говорит о длительности разработки оцененного продукта. Длительность может быть получена при наличии у организации собственной статистики длительностей на реализацию функциональных точек. Если такой статистики нет, то для оценки длительности и трудоемкости проекта можно использовать метод COSOMO или его модификации.

Методика COSOMO позволяет оценить трудоемкость и время разработки программного продукта. Впервые была опубликована Бари Боэмом [5] в 1981 году в виде результата анализа 63 проектов компании «TRW Aerospace». В 1997 году методика была усовершенствована и получила название COSOMO II. Калибровка параметров производилась по 161 проекту разработки. В модели используется формула регрессии с параметрами, определяемыми на основе отраслевых данных и характеристик конкретного проекта. Разли-

чаются две стадии оценки проекта: предварительная оценка на начальной фазе и детальная оценка после проработки архитектуры проекта.

Формула оценки трудоемкости проекта в чел. · мес. имеет вид:

$$PM = A \cdot SIZE^E \cdot \prod_{i=1}^n EM_i,$$

$$A = 2,94,$$

$$E = B + 0,01 \cdot \sum_{j=1}^5 SF_j,$$

$$B = 0,91,$$

где A – коэффициент, учитывающий мультипликативные эффекты от размера проекта; $SIZE$ – размер продукта в тысячах строк кода; E – коэффициент для учета экономии (продуктивности) проекта в зависимости от его масштаба (если $E < 1$, то продуктивность проекта увеличивается при увеличении его масштаба; если $E = 1$, то продуктивность сбалансирована, часто используется для малых проектов; если $E > 1$, то продуктивность проекта снижается при увеличении его масштаба, как правило это связано с двумя факторами: увеличением количества межличностных взаимодействий и ростом количества интеграционных механизмов); B – начальное значение для коэффициента учета экономии (E); EM_i – множители трудоемкости; SF_j – факторы масштаба, $n = 7$ – для предварительной оценки, $n = 17$ – для детальной оценки.

Главной особенностью методики является то, что для оценки трудоемкости необходимо знать размер программного продукта в тысячах строках исходного кода. Размер может быть оценен разными способами: экспертно, по аналогам, расчетным путем и т. д. Если предварительно был проведен анализ ПО, то количество строк кода может быть оценено по статистическим оценкам отрасли или проектной организации. Подробное описание метода можно найти в первоисточнике [5], либо в изложении отечественных авторов [4, 6].

Длительность проекта в методике COSOMO II рассчитывается по формуле:

$$TDEV = 3,67 \cdot (PM_{NS})^{0,28+0,2 \cdot 0,01 \cdot \sum_{j=1}^5 SF_j} \cdot \frac{SCED}{100},$$

где PM_{NS} – трудоемкость проекта без учета множителя $SCED$, определяющего сжатие расписания [7].

В отечественной практике оценки длительности или трудоемкости разработки ПО привязываются к количеству операторов или строк кода, например, по следующей формуле:

$$T_{ПО} = T_0 + T_{и} + T_a + T_{п} + T_{отл} + T_{д},$$

где T_0 – затраты труда на описание задачи; $T_{и}$ – затраты на исследование предметной области; T_a – затраты на разработку блок-схемы; $T_{п}$ – затраты на программирование; $T_{отл}$ – затраты на отладку программы; $T_{д}$ – затраты на подготовку документации. Каждое из значений затрат связано с количеством операторов или строк кода через условное количество операторов:

$$Q = q \cdot c \cdot (1 + p),$$

где q – число операторов; c – коэффициент сложности задачи (принимается от 1,25 ... 2); p – коэффициент коррекции программы, учитывающий новизну проекта (для совершенно новой программы равен 0,1).

$$T_{и} = (Q \cdot B) / (S_{и} \cdot k),$$

где B – коэффициент увеличения затрат труда вследствие недостаточного описания задачи (1,2 – 1,5); $S_{и}$ – количество операторов, приходящихся на 1 чел. · час (75–85); k – коэффициент квалификации работника.

$$T_a = Q / (S_a \cdot k),$$

где $S_a = 20$ –25.

$$T_{отл} = Q / (S_{отл} \cdot k),$$

где $S_{отл} = 4$ –5.

$$T_{д} = (1 + 0,75) \cdot Q / (S_{др} \cdot k),$$

где $S_{др} = 15$ –20; $(1 + 0,75)$ – коэффициент, учитывающий необходимость разработки и редактирования документации.

Полученное значение общей трудоемкости необходимо скорректировать с учетом уровня языка программирования:

$$T_{кор} = T \cdot k_{кор},$$

где $k_{кор}$ – коэффициент, учитывающий уровень языка программирования (0,8–1) [8].

Практика показывает, что использование собственного опыта или опыта коллег, приобретенного в похожих проектах, – это наиболее прагматичный подход, который позволяет получить наиболее вероятностные оценки длительности и трудоемкости разработки ПО, быстро и без больших затрат. Так в работах [9–11] было предложено получать длительность выполнения операций статистическим методом, причем при проведении

расчетов использование большого количества критериев, например, указывающих на опыт или квалификацию работников, тип оборудования, выполняющего данную операцию, и т. д., приводило к усложнению расчетов, а точность результата не повышалась. Исходя из данного опыта первоначальные расчеты проводились без учета качественного состава разработчиков, но с учетом структурной сложности ПО. Для учета сложности использовались следующие критерии: количество строк кода, количество операторов ветвления – if, else, for, while, repeat. Детальные результаты выбора критериев выходят за рамки данной статьи и будут представлены в другой публикации.

3 Модель оценки прогноза изготовления АПК

Пусть есть множество заказов, выполняемых проектной организацией $\{Z_i\}$. Для каждого заказа определен период его выполнения $[T_{beg}^{Z_i}, T_{end}^{Z_i}]$ (в предлагаемой работе рассматривается выполнение заказа в части разработки ПО).

Также в проектной организации есть множество подразделений $\{D_i\}$, каждое из которых работает над некоторым подмножеством заказов $\{Z_j^i\}$, периоды выполнения различных заказов могут пересекаться. Как допущение примем, что над одним заказом работает только одно подразделение, то есть $Z_k^i \cap Z_m^i = \emptyset \forall i \neq j$.

Пусть V^{Z_i} – общий объем работ по разработке ПО для заказа V^{Z_i} . Тогда в единицу времени t (день или нормочас) требуемый объем работ по заказу составит

$$V_t^{Z_i} = \frac{V^{Z_i}}{T_{end}^{Z_i} - T_{beg}^{Z_i} + 1}.$$

Если считать, что работы выполняются подразделением равномерно, то величина $V_t^{Z_i}$ должна быть обратно пропорциональна количеству заказов, выполняемых в данный период времени. Таким образом определяем коэффициент, обратно пропорциональный загруженности подразделения по всем заказам в данный период времени t и определяющий ту часть трудоемкости, которую может выделить подразделение на конкретный заказ:

$$K_t = \frac{V_t^{Z_i}}{\sum_j V_t^{Z_j}}.$$

Таким образом, можно определить реальный объем работ, который может выполнить подразделение по заказу Z_i в период времени t_k из расчета на одного сотрудника, должность которого подходит для выполнения работ:

$$V_{t_k}^{Z_i} = V_t^{Z_i} \cdot K_t.$$

Также необходимо учесть количество сотрудников, работающих в подразделении в данный период времени и по должности подходящих для выполнения работ по заказу.

Пусть C_0^i – общее количество сотрудников подразделения D_i , имеющих подходящую должность; C^i – количество реально работающих в данный период времени (не в отпусках, не на больничных). Тогда коэффициент работоспособности подразделения определяется следующим образом:

$$K_{\text{раб}}^i = \frac{C^i}{C_0^i}.$$

Окончательный расчет реального объема выполняемых работ по заказу Z_i в период времени t определяется по формуле:

$$V_{t_k}^{Z_i} = V_t^{Z_i} \cdot K_t \cdot C^i.$$

Таким образом, имея данные по объему работ на каждом заказе, количество трудовых ресурсов подразделений и временные рамки выполнения заказов, можно определить реальность выполнения заданного объема работ и выдать рекомендации по изменению сроков, численности подразделений и перераспределению работ между подразделениями.

Далее стоит отметить определение трудоемкости выполнения работ, исходя из количества разработанных строк кода.

Пусть $V_{\text{план}}$ – плановая трудоемкость выполнения работ по заказу, $V_{\text{факт}}$ – фактическое количество дней, затраченных на выполнение работ, C_{all} – количество строк кода всех исходных текстов, разработанных в рамках заказа. Тогда получаем значения плановой и фактической трудоемкости разработки одной строки кода:

$$T_{\text{план}} = \frac{V_{\text{план}}}{C_{\text{all}}}, \quad T_{\text{факт}} = \frac{V_{\text{факт}}}{C_{\text{all}}}.$$

Естественно, что при расчетах по статистическим данным может быть получено некоторое значение ошибки $|T_{\text{план}} - T_{\text{факт}}|$. Для более точного прогнозирования трудоемкости разработки необходимо минимизировать значение ошибки, для чего возможно воспользоваться таким понятием, как сложность разрабатываемого кода (алгоритма). Понятно, что для статистических оценок невозможно использовать реальные виды алгоритмов и их сложность, так как в явном виде они не указаны. Однако есть возможность оценить примерную сложность, используя управляющие структуры и операторы, присутствующие в коде (например, операторы условного перехода, циклы и т. д.). В структуре базы данных (БД) (рис. 5) уже учитываются данные по количеству подобных операторов, полученные при разборе файлов исходных текстов ПО. Примем за коэффициент коррекции ошибки

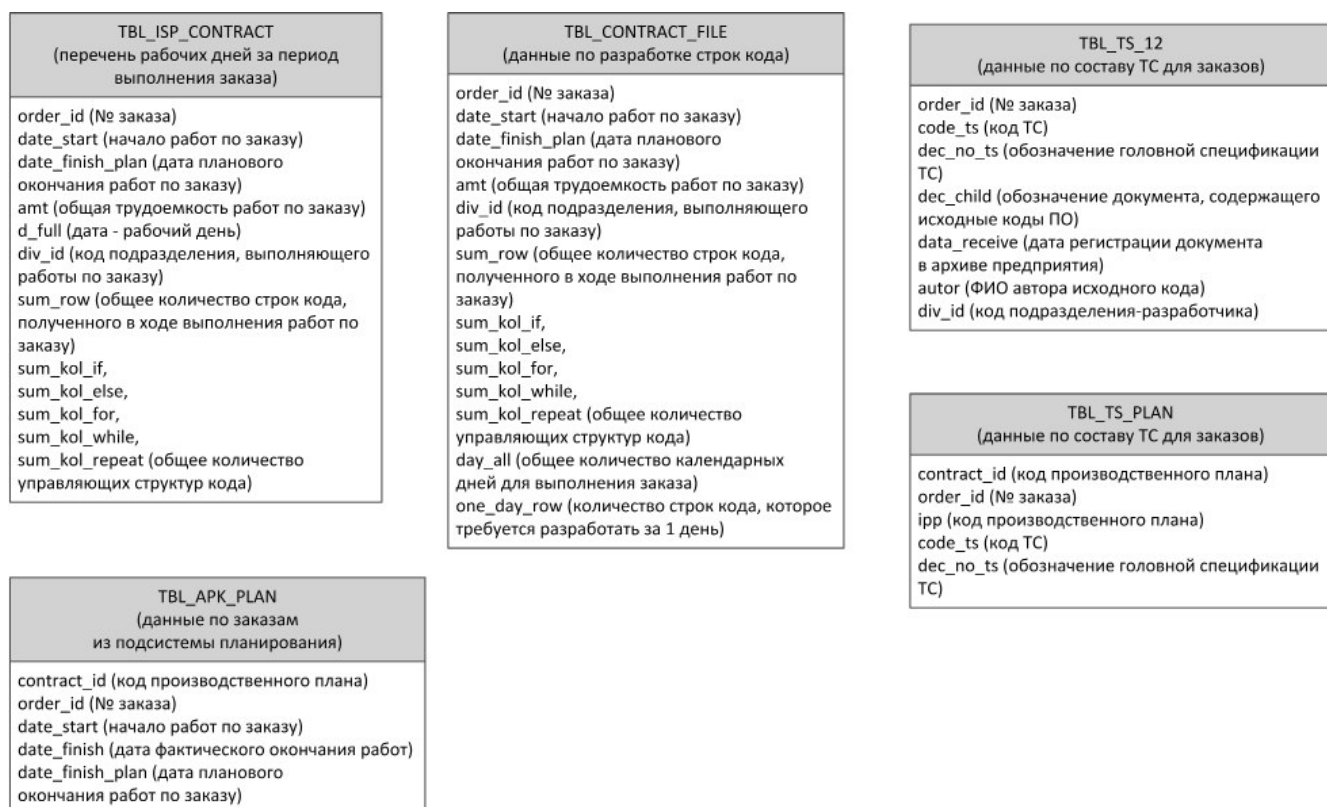


Рис. 5. Структура БД для оценки длительности изготовления ПО

Требуемые и реальные затраты на выполнение работ по заказам

№ заказа	Код подразделения	Требуемый объем работ	Реальный объем работ
01182222	0213	2050,28	3325,000000
00738211	0131	1654	1101,712682
00495211	0131	13182	13127,658474
00745211	0131	3000	1834,373926
00705211	0131	3000	1902,255702

$$K_{corr} = K_{if} \cdot C_{if} + K_{else} \cdot C_{else} + K_{for} \cdot C_{for} + K_{while} \cdot C_{while} + K_{repeat} \cdot C_{repeat},$$

где $K_{опер}$ – коэффициент значимости оператора «опер», который принимает значение из множества {if, else, for, while, repeat},

$C_{опер}$ – количество операторов данного вида, зафиксированных в исходном тексте ПО.

Вычисление значений коэффициентов, имея достаточно статистических данных, можно произвести с помощью простой нейронной сети [12–14] и в дальнейшем использовать при прогнозе сроков разработки ПО.

Общая формула для расчета прогнозируемой трудоемкости разработки строки кода имеет вид:

$$T_{план} = K_{corr} \cdot \frac{V_{план}}{C_{all}}$$

Поставлена задача: на основе статистических данных за некоторый прошедший период оценить указанную трудоемкость и сравнить ее с результатами оценки, полученными с помощью разработанного алгоритма. Для оценки ошибки предлагается относительное значение

$$\varepsilon = \frac{|T_{план} - T_{факт}|}{T_{план}} \cdot 100\%.$$

Для реализации предлагаемого алгоритма сформирована БД, в которой были собраны данные о ходе планов разработки ПО, о составе изделий, о ПО. Структура приведена на рисунке 5.

4 РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Исходя из приведенной структуры данных и модели оценки прогноза, были получены следующие расчетные данные (пример выборки приведен в таблице 1).

Из приведенной выборки видно, что можно дать рекомендации перераспределить объем выполняемых работ с наиболее загруженных на наименее загруженные подразделения.

Расчет коэффициентов важности операторов не представляет собой достаточно сложную задачу и выходит за рамки предлагаемой работы, результат расчета представлен в таблице 2 (в расчет приняты наиболее часто встречающиеся в исследуемом коде операторы условного перехода и циклов).

Таблица 2

Коэффициенты важности операторов

оператор	if	else	for	while	repeat
коэффициент	0,45	0,3	0,6	0,4	0,2

Пример расчета трудоемкости разработки кода приведен в таблице 3.

По результатам экспериментов видно, что значение ошибки не превышает 30%.

ЗАКЛЮЧЕНИЕ

Использование ранее накопленного опыта позволяет существенно увеличить точность оценки длительно-

Таблица 3

Пример расчета трудоемкости разработки строки исходного кода ПО

заказ	дата нач.	дата оконч.	трудоемкость план	кол-во строк	кол-во if	кол-во else	кол-во for	кол-во while	кол-во repeat	кол-во дней	трудоемкость строки факт	трудоемкость строки план	ε, %
00495211	18.09.15	30.03.21	13182	180125	12854	1423	6165	1095	1201	2020	0,015058	0,011214446	26
00705211	09.03.17	30.06.19	3000	90128	6854	823	4166	690	211	843	0,01106	0,009353373	15
00738211	05.12.17	20.02.20	1654	11256	2451	681	2563	2075	547	807	0,066863	0,071695168	7
00745211	05.05.17	30.09.19	3000	90150	6892	858	4201	705	315	878	0,013969	0,009739333	30
01182222	01.01.20	03.12.21	2050,28	27516	1784	287	435	28	1	702	0,034134	0,025512455	25
04303189	17.01.17	15.02.19	1200	9946	150	73	274	13	0	759	0,061272	0,076312162	25

сти и трудоемкости разработки ПО для сложных АПК. Особенную ценность представляет формализованный опыт – формализованные показатели производственного процесса, которые позволяют непрерывно сравнивать производственный процесс организации в прошлом, настоящем и давать оценки на будущее.

В статье представлены методика оценки длительности и трудоемкости разработки ПО с использованием формализованного опыта предыдущих периодов, механизмы извлечения формальных показателей для проведения оценивания.

Основным недостатком предложенного подхода являются ограничения, связанные с наличием какого-либо ПО, разрабатываемого в предыдущие периоды. В то же время переход от неформализованного метода оценки сроков разработки ПО к механизмам оценки сроков с помощью предложенной методики является существенным улучшением, т. к. позволяет проводить оценку сроков изготовления в кратчайшие сроки с учетом формальных показателей, присущих конкретным изделиям.

Таким образом, следует сказать, что предложенный механизм позволяет повысить точность предварительной оценки длительности разработки ПО без существенных доработок используемой системы автоматизации планирования производственной деятельности.

СПИСОК ЛИТЕРАТУРЫ

1. Радионова Ю.А. Интеллектуальные методы организации архивов технической документации научно-производственного объединения : дис. ... канд. техн. наук. Ульяновск, 2012.

2. ГОСТ 2.501-2013. Межгосударственный стандарт. Единая система конструкторской документации. Правила учета и хранения. URL: <https://docs.cntd.ru/document/1200106864> (дата обращения: 23.07.2021).

3. Albrecht A.J., Gaffney J.E. Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation // Proc. IEEE Transactions on Software Engineering, IEEE Press, Nov. 1983. pp. 639–648.

4. Архипенков С. Лекции по управлению программными проектами. URL: http://www.arkhipenkov.ru/resources/sw_project_management.pdf (дата обращения: 11.08.2021).

5. Boehm B.W. Software Engineering Economics. //IEEE Transactions on Software Engineering, 1984, vol. SE-10, no. 1, pp. 4–21. Available at, 1994. URL: <http://www.pauldee.org/se-must-have/boehm-SE-Economics.pdf> (дата обращения: 11.08.2021).

6. Шанченко Н.И. Оценка трудоемкости разработки программного продукта: метод. указания. Ульяновск : УлГТУ, 2015. 40 с.

7. Boehm B.W. et al. An Overview of COCOMO 2.0 Software Cost Model. URL: <https://www.researchgate.net/publication/239401864> (дата обращения: 23.07.2021).

8. Пример расчета трудоемкости разработки программного продукта. URL: <https://lektcii.net/3-49502.html> (дата обращения: 11.08.2021).

9. Перцев А.А., Подобрый А.Н., Радионова Ю.А. Планирование расхода материалов для обеспечения производства машиностроительного предприятия // Автоматизация процессов управления. 2021. № 1 (63). С. 34–45.

10. Перцев А.А., Подобрый А.Н. Подход к автоматизированному формированию плана изготовления производственного подразделения проектной организации // Сб. науч. тр. 7-й Всерос. науч.-техн. конф. аспирантов, студентов и молодых ученых «Информатика и вычислительная техника» (ИВТ-2015), г. Ульяновск, 25–27 мая 2015. Ульяновск : УлГТУ, 2015. С. 398–405.

11. Перцев А.А., Подобрый А.Н., Радионова Ю.А. Реализация равномерной загрузки производства машиностроительного предприятия // Автоматизация процессов управления. 2020. № 4 (62). С. 49–60.

12. Mrinal Kanti Ghose, Roheet Bhatnagar, Vandana Bhattachajee. Comparing Some Neural Network Models for Software Development Effort Prediction. // Paper of the 2nd National Conference on Emerging Trends and Applications in Computer Science (NCETACS). March 2011. URL: <https://researchgate.net/publication/236868569> (дата обращения: 23.07.2021).

13. Pichai Jodpimai, Peraphon Sophatsathit and Chidchanok Lursinsap. Estimating Software Effort with Minimum Features using Neural Functional Approximation // IEEE Xplore. Fukuoka, Japan, Mai, 2010. DOI:10.1109/ICCSA. URL: <https://researchgate.net/publication/228955154> (дата обращения: 23.07.2021).

14. Anupama Kaushik, Ashish Chauhan, Sachin Gupta, Deepak Mittal. COCOMO Estimates Using Neural Networks // Int. J. Intelligent Systems and Applications, 2012, iss. 9, pp. 22–28. URL: <https://researchgate.net/publication/266889462> (дата обращения: 23.07.2021).

REFERENCES

1. Radionova Ju.A. *Intellektualnye metody organizatsii arkhivov tekhnicheskoi dokumentatsii nauchno-proizvodstvennogo obedineniia*. Dis. kand. tekhn. Nauk [Intelligent Methods of Organization of Technical Documentation Archive in a Research and Production Centre. Abstract of a Candidate of Eng. Sci. Diss.]. Ulyanovsk, 2012.

2. *GOST 2.501–2013. Mezhhgosudarstvennyi standart. Edinaia sistema konstruktorskoj dokumentatsii. Pravila ucheta i khraneniia* [State Standard 2.501–2013. Unified System for Design Documentation. Registration and Storage Rules]. Available at: <https://docs.cntd.ru/document/1200106864> (accessed 23.07.2021).

3. Albrecht A.J., Gaffney J.E. Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. *Proc. IEEE Transactions on Software Engineering*. IEEE Press, Nov. 1983, pp. 639–648.

4. Arkhipenkov S. *Lektsii po upravleniiu programmnyimi proektami* [Lectures on Software Project Management]. Available at: http://arkhipenkov.ru/resources/sw_project_management.pdf (accessed 11.08.2021).

5. Boehm B.W. Software Engineering Economics. *IEEE Transactions on Software Engineering*, 1984, vol. SE-10,

- no. 1, pp. 4–21. Available at: <http://pauldee.org/se-must-have/boehm-SE-Economics.pdf> (accessed 11.08.2021).
6. Shanchenko N.I. *Otsenka trudoemkosti razrabotki programmnogo produkta. Metod. ukazaniia* [Evaluation of Labor Intensity of Software Product Development. Methodological Guidance]. Ulyanovsk, UISTU Publ., 2015. 40 p.
7. Boehm B.W. et al. *An Overview of COCOMO 2.0 Software Cost Model*. Available at: <https://researchgate.net/publication/239401864> (accessed 23.07.2021)
8. *Primer rascheta trudoemkosti razrabotki programmnogo produkta* [An Example of Calculating the Labor Intensity of Software Product Development]. Available at: <https://leksii.net/3-49502.html> (accessed 11.08.2021).
9. Pertsev A.A., Podobrii A.N., Radionova Iu.A. Planirovanie raskhoda materialov dlia obespecheniia proizvodstva mashinostroitel'nogo predpriiatiia [Material Scheduling to Provide Manufacturing in Machine Industry]. *Avtomatizatsiia protsessov upravleniia* [Automation of Control Processes], 2021, no. 1 (63), pp. 34–45.
10. Pertsev A.A., Podobrii A.N. Podkhod k avtomatizirovannomu formirovaniu plana izgotovleniia proizvodstvennogo podrazdeleniia proektnoi organizatsii [An Approach to Automated Production Planning in a Manufacturing Unit of Project Organization]. *Sb. nauch. tr. 7-i Vseros. nauch.-tekhn. konf. aspirantov, studentov i molodykh uchenykh "Informatika i vychislitelnaia tekhnika" (IVT-2015)* [Proc. of the 7th Russ. Sci.- Tech. Conf. for Postgraduates, Students and Young Scientists on Informatics and Computer Facilities]. Ulyanovsk, UISTU Publ., 2015, pp. 398–405.
11. Pertsev A.A., Podobrii A.N., Radionova Iu.A. Realizatsiia ravnomernoi zagruzki proizvodstva mashinostroitel'nogo predpriiatiia [The Balanced Production Load of a Machine-Engineering Organization]. *Avtomatizatsiia protsessov upravleniia* [Automation of Control Processes], 2020, no. 4 (62), pp. 49–60.
12. Mrinal Kanti Ghose, Roheet Bhatnagar, Vandana Bhattachajee. Comparing Some Neural Network Models for Software Development Effort Prediction. *Paper of the 2nd National Conference on Emerging Trends and Applications in Computer Science (NCETACS)*. March 2011. Available at: <https://researchgate.net/publication/236868569> (accessed 23.07.2021).
13. Pichai Jodpimai, Peraphon Sophatsathit and Chidchanok Lursinsap. Estimating Software Effort with Minimum Features using Neural Functional Approximation. *IEEE Xplore*. Fukuoka, Japan, Mai, 2010. DOI:10.1109/ICCSA. Available at: <https://researchgate.net/publication/228955154> (accessed 23.07.2021).
14. Anupama Kaushik, Ashish Chauhan, Sachin Gupta, Deepak Mittal. COCOMO Estimates Using Neural Networks. *Int. J. Intelligent Systems and Applications*, 2012, iss. 9, pp. 22–28. Available at: <https://researchgate.net/publication/266889462> (accessed 23.07.2021).