

МЕТОД РАСЧЕТА ГАРАНТИРОВАННОГО ВРЕМЕНИ МОДИФИКАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

А.А. Байбулатов

Предложен метод расчета гарантированного (максимального) времени модификации программного обеспечения с длительным итерационным процессом разработки на основе аппарата «Network calculus». Выполнены расчеты максимального времени модификации программного обеспечения АСУТП АЭС на этапах пуска-наладки и эксплуатации для различных способов модификации.

Ключевые слова: время модификации, программное обеспечение, «Network calculus», АЭС.

ВВЕДЕНИЕ

При разработке и сопровождении программного обеспечения (ПО) нередко возникает задача предварительной оценки временных затрат на его разработку [1]. Особый интерес представляет не полное время разработки проекта (которое определяется заказчиком и разработчиком на основе их квалификации и опытных данных и фиксируется в соответствующих документах на начальном этапе жизненного цикла ПО), а время разработки промежуточных работоспособных версий, т. е. время модификации. Решение этой задачи необходимо для планирования сроков разработки (модификации), а также для привлечения необходимого количества трудовых ресурсов. Особенно актуальна эта задача для промышленных систем с высоким экономическим риском в случае простоя и объектов повышенного риска эксплуатации. В настоящее время имеется ряд моделей и методов для прогнозирования временных затрат на разработку (модификацию) ПО.

Структурно-логическая модель прогнозирования времени разработки ПО [2] дает наглядное представление «сверху — вниз» на качественном уровне, но для получения с ее помощью количественных оценок необходимы экспертные данные [3], которые не всегда можно считать корректными [1]. Технология нечеткого прогнозирования [4] основана на методе нечеткого логического вывода. Отдельно можно выделить методы оценки, основанные на аналогии [5], которые работают совместно с нечеткой логикой, генетическими алго-

ритмами и др. Однако все перечисленные методы не учитывают итерационный характер разработки ПО, которого придерживается большинство современных разработчиков, а ориентированы, в основном, на «водопадную» модель [6].

Для оценки характеристик итерационного процесса разработки ПО — временных затрат на одну итерацию разработки или версию ПО необходимы другие методы. В большинстве случаев для решения подобных задач применяется классическая теория массового обслуживания [7]. В некоторых случаях применяются также ее более современные развитые формы. Например, многофазные системы массового обслуживания [8], которые состоят из нескольких типовых узлов, расположенных последовательно, т. е. представляют собой совокупность нескольких систем массового обслуживания. Тем не менее, любая самая сложная модель теории массового обслуживания работает с понятиями теории вероятностей и может предсказать только статистически среднее значение определяемой величины. Но известно, что статистическими характеристиками трудно воспользоваться: для дальнейших расчетов и планов, как правило, нужны характеристики детерминированные. Кроме того, средние статистические характеристики совершенно не подходят для промышленных систем с высоким экономическим риском в случае простоя и объектов повышенного риска эксплуатации, где наибольший интерес представляют наихудшие значения прогнозируемых величин. При оценке временных затрат для таких систем необходимо предсказать не среднее время разработки, а гарантированное (максимальное).



Предлагаемый метод на основе аппарата «Network calculus» позволяет достаточно легко решить задачу оценки максимального времени разработки версии ПО (модификации ПО).

В работе исследована применимость аппарата «Network calculus» для расчета максимального времени модификации программного обеспечения АСУТП АЭС разработки Института проблем управления им. В.А. Трапезникова РАН [9] версии для АЭС «Куданкулам» (Индия). Оценка максимального времени модификации ПО представляет собой важную задачу, поскольку сроки модификации могут быть жестко ограничены временем планово-предупредительного ремонта или внепланового останова энергоблока АЭС. Под ПО в работе понимается его часть в виде информационной базы сигналов и графических мнемосхем.

1. ЗАДАЧА РАСЧЕТА ГАРАНТИРОВАННОГО ВРЕМЕНИ МОДИФИКАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

1.1. Аппарат «Network Calculus»

«Network calculus» [10] — метод анализа детерминированных систем с очередью, который применяется для расчета характеристик вычислительных сетей [11–13] и оперирует понятиями мини-плюс алгебры [14]. Основы метода заложены в конце прошлого века [15–18].

Аппарат метода «Network calculus» может быть также применен для решения задачи расчета гарантированного (максимального) времени разработки версии ПО или модификации ПО. Проводя аналогию с вычислительными сетями, может быть оценена «задержка» разработки, т. е. время обработки одного пакета заданий при заданном потоке.

Задача может быть поставлена следующим образом. Задания на разработку (модификацию) ПО имеют вид потока. Разработка (модификация) ПО имеет итерационный характер. Необходимо оценить максимальное время, за которое может быть проведена модификация ПО. Особенность задачи состоит в том, что необходимо оценить не полное время разработки ПО, а время модификации, которое, как правило, соответствует одной итерации разработки с учетом очереди входного потока.

Для решения этой задачи потребуются некоторые понятия аппарата «Network calculus».

Функция потока $F(t)$ — неотрицательная неубывающая функция времени.

Огибающая потока $\alpha(t)$ — возрастающая функция, ограничивающая функцию потока (рис. 1).

Доказано [10], что:

$$\alpha(t) = \sup_{\tau \geq 0} \{F(t + \tau) - F(\tau)\}. \quad (1)$$

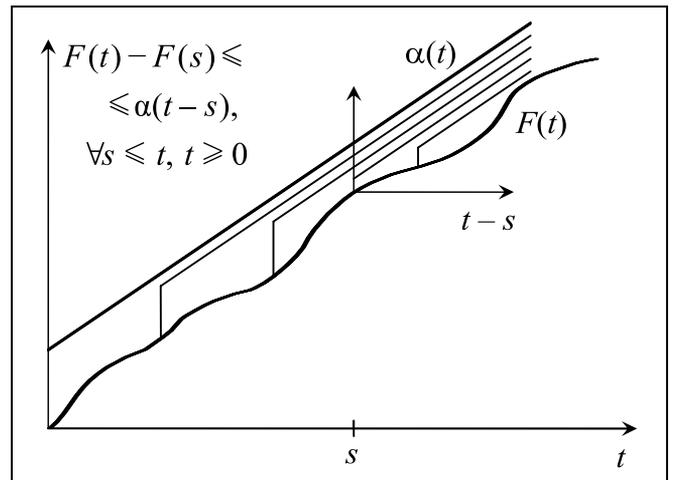


Рис. 1. Функция потока и его огибающая

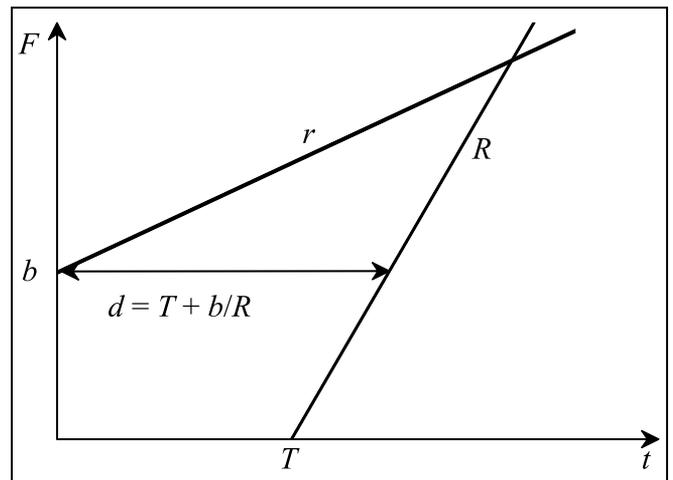


Рис. 2. Максимальное время модификации для однокомпонентной модели

Функция обслуживания:

$$\beta(t) = R(t - T), \quad t > T,$$

$$\beta(t) = 0, \quad t \leq T,$$

где R — скорость обслуживания, T — задержка обслуживания.

Рассмотрим две модели метода «Network Calculus», отличающиеся описанием входного потока.

Однокомпонентная модель. Огибающая входного потока имеет вид $\alpha(t) = rt + b$, где r — скорость, b — максимальный размер пакета.

Максимальное время модификации [10] (рис. 2):

$$d = b/R + T, \quad r \leq R. \quad (2)$$

Двухкомпонентная модель IntServ. Огибающая входного потока имеет вид $\alpha(t) = \min(r_1 t + b_1, r_2 t + b_2)$, где r_1 и b_1 , r_2 и b_2 — скорость и макси-

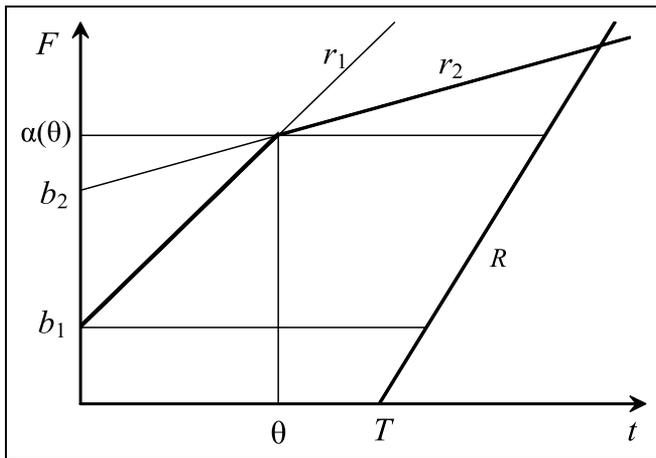


Рис. 3. Максимальное время модификации для двухкомпонентной модели IntServ

мальный размер пакета для первой и второй компонент соответственно.

Условие применимости двухкомпонентной модели IntServ: $b_1 < b_2$, $r_1 > r_2$.

Максимальное время модификации [10] (рис. 3):

$$d = \max(\alpha(\theta)/R + T - \theta, b_1/R + T), \quad (3)$$

где $\theta = \frac{b_2 - b_1}{r_1 - r_2}$.

Для расчета максимального времени модификации ПО с помощью аппарата «Network calculus» необходимы характеристики входного потока (скорость и максимальный размер пакета или огибающая) и функции обслуживания (разработки ПО).

Очевидно, что условием применимости аппарата «Network calculus» является итерационный характер процесса разработки: задания на разработку должны описываться потоком. При этом скорость обслуживания (разработки) должна превышать скорость входного потока (потока заданий).

1.2. Алгоритм решения

Схема обобщенного алгоритма расчета максимального времени модификации ПО приведена на рис. 4. Рассмотрим его более подробно.

Шаг 1. Выбор модели (одно- или двухкомпонентной). При выборе модели можно руководствоваться:

- видом входного потока (если он известен),
- типом модификации ПО.

Нередко внешний вид входного потока в графическом кумулятивном представлении за некоторый промежуток времени из прошлого может помочь выбрать модель. В некоторых случаях легко определяется «точка перегиба», которая разделяет поток на две компоненты (см. далее рис. 6).

Модификацию ПО, в зависимости от причин, можно разделить на два типа: устранение ошибок и добавление новых функциональных возможностей, ввод новых модулей, подсистем и т. п. В случае, если присутствует модификация (и соответственно входной поток) только одного типа, модель можно выбрать однокомпонентную; при наличии обоих типов модификации (и входных потоков) лучше подойдет модель двухкомпонентная.

Если рассмотреть жизненный цикл промышленной системы, для которой разрабатывается и модифицируется ПО, то можно отметить, что на этапе пуско-наладки присутствуют оба типа модификации, поэтому в этом случае имеет смысл воспользоваться двухкомпонентной моделью; на этапе эксплуатации, как правило, модификация определяется только устранением ошибок, поэтому подойдет однокомпонентная модель.

Шаг 2. Определение характеристик входного потока. Аппарат «Network calculus» не накладывает никаких ограничений на входной поток — поток может иметь произвольные характеристики. Однако применение данного аппарата нецелесообразно в двух случаях: при сильно разреженном входном потоке, когда результат очевиден; при нестабильном входном потоке. Применение «Network calculus» оправдано в случае потока со стабильными характеристиками. При модификации, связанной с устранением ошибок, входной поток определяется оставшимися в ПО ошибками, которые, как известно, хорошо описываются моделями, в которых экспоненциальная зависимость количества ошибок от времени на длительных временных интервалах переходит в константу [19] и соответственно стабилизирует входной поток. Применять «Network calculus» желательно именно на этих временных интервалах.

Если входной поток за некоторый промежуток времени из прошлого известен, то характеристики потока можно определить на основе этого временного промежутка. Если входной поток за прошедшее время неизвестен, то характеристики потока можно установить экспертным путем. Например, максимальный размер пакета заданий на разработку можно принять равным максимальному размеру проекта ПО.



Рис. 4. Обобщенный алгоритм расчета максимального времени модификации ПО



Шаг 3. Вывод функций обслуживания. Функции обслуживания всегда можно найти экспериментальным путем посредством исследования зависимости времени разработки ПО от объема заданий на разработку.

Шаг 4. Численные расчеты. После определения характеристик входного потока и функций обслуживания нетрудно провести численные расчеты по формуле (2) для однокомпонентной модели и по формуле (3) — для двухкомпонентной.

2. РАСЧЕТ ГАРАНТИРОВАННОГО ВРЕМЕНИ МОДИФИКАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1. Постановка задачи

Применим (исследуем применимость) изложенный метод на основе аппарата «Network calculus» для расчета гарантированного (максимального) времени модификации программного обеспечения АСУТП первого энергоблока АЭС «Куданкулам» (Индия).

Задачу можно поставить следующим образом. Пусть модификация ПО осуществляется одним проектировщиком после получения каждого пакета заданий; поток заданий на модификацию за прошедшее время известен. Необходимо оценить максимальное время, за которое может быть проведена модификация ПО одним проектировщиком в будущем.

2.2. Анализ входного потока

Рассмотрим поток заданий на мнемосхемы в жизненном цикле АСУТП АЭС (график потока приведен на рис. 5, его характеристики — в табл. 1).

Для потока на этапе эксплуатации будем пользоваться однокомпонентной моделью, а на этапе пуско-наладки — двухкомпонентной.

Для нахождения параметров двухкомпонентной модели преобразуем поток заданий на мнемосхемы на этапе пуско-наладки к виду с нарастающей суммой (кумулятивному виду) (рис. 6).

Приведем временную шкалу к равномерной сетке и сделаем пересчет на рабочие дни. Выполним аппроксимацию потока методом наименьших квадратов линейными функциями. По формуле (1) численными методами найдем *огibaющую потока*

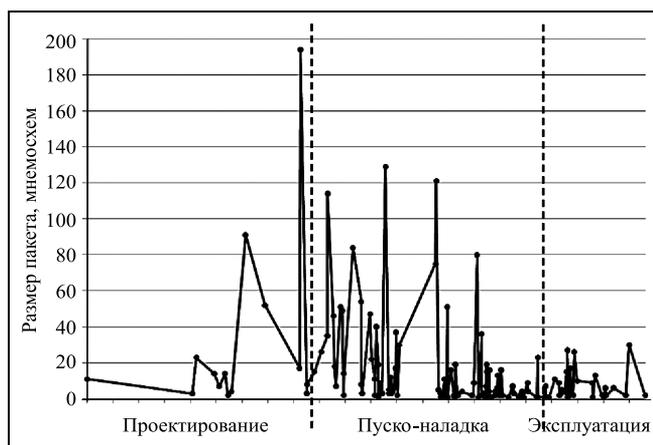


Рис. 5. Поток заданий на мнемосхемы в жизненном цикле АСУТП АЭС

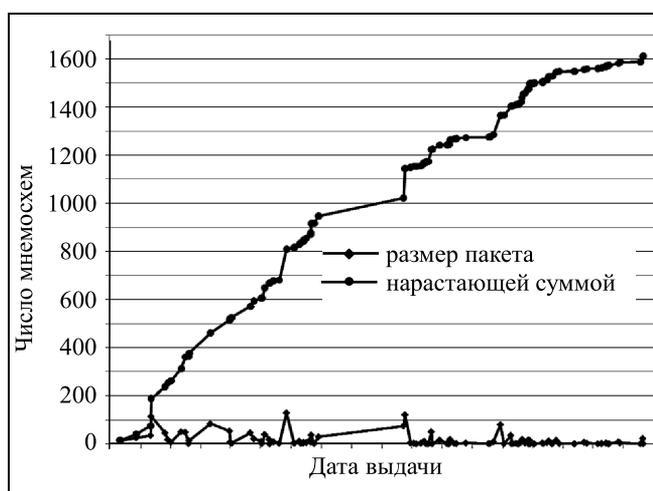


Рис. 6. Поток заданий на мнемосхемы на этапе пуско-наладки (пакетный вид и вид с нарастающей суммой)

и также аппроксимируем ее линейными функциями (рис. 7):

$$N_1 = 3,13t + 90, \quad N_2 = 1,65t + 530. \quad (4)$$

Для этапа эксплуатации (однокомпонентной модели) *огibaющую потока* искать не будем, а вос-

Таблица 1

Характеристики потока заданий на мнемосхемы в жизненном цикле АСУТП АЭС

Этап	Продолжительность, раб. дней	Средний размер пакета, мнемосхем	Среднеквадратическое отклонение	Число пакетов	Число пакетов/раб. день	Максимальный размер пакета, мнемосхем
Проектирование	751	31,64	52,69	14	0,0186	194
Пуско-наладка	782	15,49	25,37	104	0,1330	129
Эксплуатация	348	7,24	7,77	34	0,0977	27

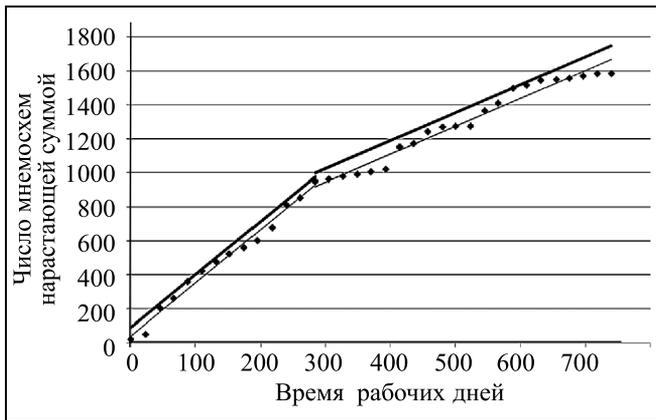


Рис. 7. Поток заданий на мнемосхемы на этапе пуско-наладки АСУТП АЭС (нарастающей суммой с приведением временной шкалы к равномерной сетке и пересчетом на рабочие дни): тонкие линии — аппроксимированный поток мнемосхем, жирные линии — аппроксимированная огибающая потока

пользуемся экспериментально полученными характеристиками (см. табл. 1): максимальный размер пакета $b = 27$, скорость найдем по формуле

(средний размер пакета) · (число пакетов/раб. день):

$$r = 7,24 \cdot 0,0977 = 0,71.$$

2.3. Вывод функций обслуживания

Функции обслуживания найдем экспериментальным путем. Для этого рассмотрим цикл разработки (модификации) ПО (рис. 8): получение задания на разработку (модификацию), разработка (модификация), тестирование, поставка. Будем считать, что тестирование проходит успешно и никаких коррекций вносить не требуется.

Определим несколько способов разработки (модификации) ПО (далее в тексте указаны подстрочными индексами):

auto — модификация с автоматическим созданием базы сигналов, без изменения мнемосхем в графическом редакторе;

auto, edit — модификация с автоматическим созданием базы сигналов и с изменением мнемосхем в графическом редакторе;

DB — модификация с автоматизированным («ручным») созданием базы сигналов, без изменения мнемосхем в графическом редакторе;

DB, edit — модификация с автоматизированным («ручным») созданием базы сигналов и с изменением мнемосхем в графическом редакторе.

Для этапа пуско-наладки рассмотрим только способы разработки без изменения мнемосхем, поскольку в этом случае эскизы мнемосхем приходят в заданиях в готовом виде и не нуждаются в

корректировках. Для этапа эксплуатации рассмотрим все вышеперечисленные способы разработки.

За единицу изменений примем 1 мнемосхему и найдем экспериментальную зависимость времени модификации ПО одним проектировщиком от числа мнемосхем. Будем считать, что время модификации включает в себя интервалы (состав зависит от способа разработки): время подготовки, время модификации мнемосхем в графическом редакторе, время работы с мнемосхемами в автоматизированной программе, время работы с базой сигналов в автоматизированной программе, время выполнения автоматических процедур создания базы сигналов, время выполнения автоматической процедуры объединения, время подготовки к тестированию, время тестирования, время поставки. В табл. 2 и 3 приведены усредненные времена (t) модификации программного обеспечения АСУТП АЭС в зависимости от числа мнемосхем (N) на этапах пуско-наладки и эксплуатации соответственно.

Полученные зависимости времен модификации ПО от числа мнемосхем для этапов пуско-наладки и эксплуатации представлены на графиках (рис. 9 и 10 соответственно), где они аппроксимированы методом наименьших квадратов линейными функциями, которые являются функциями обслуживания.

Таким образом, функции обслуживания для этапа пуско-наладки:

$$N_{auto} = 52,31(t - 0,24),$$

$$N_{DB} = 34,35(t - 0,22); \quad (5)$$

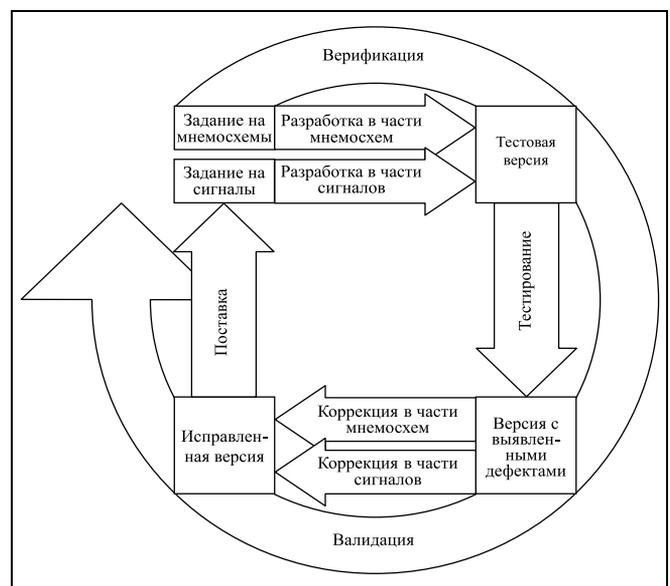


Рис. 8. Цикл разработки (модификации) ПО



для этапа эксплуатации:

$$\begin{aligned} N_{auto} &= 52,31(t - 0,12), \\ N_{auto, edit} &= 33,39(t - 0,12), \\ N_{DB} &= 34,35(t - 0,09), \\ N_{DB, edit} &= 25,03(t - 0,09). \end{aligned} \quad (6)$$

2.4. Численные результаты

Для расчета максимального времени модификации программного обеспечения АСУТП первого энергоблока АЭС «Куданкулам» на этапе эксплуатации применим однокомпонентную модель. Характеристики входного потока приведены в п. 2.2, функции обслуживания — в формулах (6). Максимальные времена модификации ПО, вычисленные по формуле (2):

$$\begin{aligned} d_{auto} &= 0,64 \text{ раб. дн.}, & d_{auto, edit} &= 0,93 \text{ раб. дн.}, \\ d_{DB} &= 0,88 \text{ раб. дн.}, & d_{DB, edit} &= 1,17 \text{ раб. дн.} \end{aligned}$$

Для расчета максимального времени модификации ПО на этапе пуско-наладки применим двухкомпонентную модель. Огибающая входного потока и функции обслуживания описываются формулами (4) и (5) соответственно. Максимальные времена модификации ПО, вычисленные по формуле (3):

$$d_{auto} = 1,96 \text{ раб. дн.}, \quad d_{DB} = 2,84 \text{ раб. дн.}$$

Отметим, что полученные численные значения полностью подтвердились фактическими работами по модификации программного обеспечения АСУТП первого энергоблока АЭС «Куданкулам», время которой, как правило, не превышало 0,7 от этих значений.

Подобный расчет максимального времени модификации ПО был проведен также для второго энергоблока. Было вычислено максимальное вре-

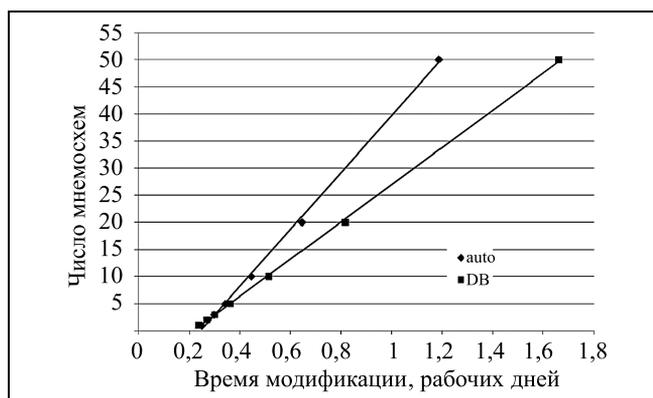


Рис. 9. Зависимость времени модификации ПО от числа мнемосхем на этапе пуско-наладки

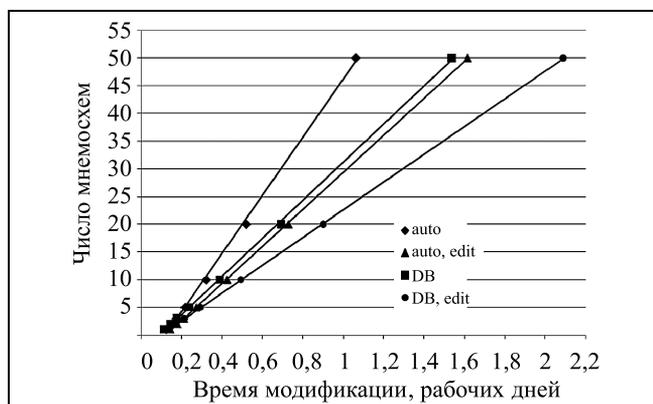


Рис. 10. Зависимость времени модификации ПО от числа мнемосхем на этапе эксплуатации

мя модификации ПО для установившегося потока первых шести месяцев этапа пуско-наладки ($d_{auto} = 0,7$ раб. дн., $d_{DB} = 0,9$ раб. дн.). На протяжении последующего времени (более одного го-

Таблица 2

Зависимость времени модификации ПО от числа мнемосхем на этапе пуско-наладки

t_{auto} , раб. дн.	0,2479	0,2729	0,2979	0,3417	0,4458	0,6458	1,1875
t_{DB} , раб. дн.	0,2354	0,2688	0,3	0,3625	0,5146	0,8167	1,6625
N	1	2	3	5	10	20	50

Таблица 3

Зависимость времени модификации ПО от числа мнемосхем на этапе эксплуатации

t_{auto} , раб. дн.	0,1229	0,1479	0,1729	0,2167	0,3208	0,5208	1,0625
$t_{auto, edit}$, раб. дн.	0,1438	0,1792	0,2104	0,2729	0,425	0,7292	1,6146
t_{DB} , раб. дн.	0,1104	0,1438	0,175	0,2375	0,3896	0,6917	1,5375
$t_{DB, edit}$, раб. дн.	0,1313	0,175	0,2125	0,2938	0,4938	0,9	2,0896
N	1	2	3	5	10	20	50

да) полученные значения успешно использовались для прогноза времени модификации ПО. Фактическое время модификации не превышало 0,7 от расчетного.

2.5. Выводы

Результаты, полученные для ПО (в части информационной базы сигналов и графических мнемосхем) АСУТП АЭС «Куданкулам», действительны также для программного обеспечения АСУТП других АЭС. Экспериментально установлено, что процесс модификации ПО характеризуется высокой скоростью обслуживания (R), небольшой задержкой обслуживания (T) и невысокой скоростью входного потока (r). Максимальное время модификации в этом случае всегда вычисляется по формуле (2), а двухкомпонентная модель вырождается в однокомпонентную. Из опыта стало также известно, что на этапе проектирования АСУТП АЭС входной поток сильно разрежен, поэтому результат очевиден, и применение аппарата «Network calculus» нецелесообразно. На этапах пуска-наладки и эксплуатации поток имеет более стабильные характеристики, и расчет максимального времени модификации может быть проведен. При этом в соответствии с экспериментальными данными, параметры максимального размера пакета (b) следует считать равными 0,6 и 0,2 от максимального числа мнемосхем в проектах информационной базы для этапов пуска-наладки и эксплуатации соответственно.

ЗАКЛЮЧЕНИЕ

Предложен метод расчета гарантированного (максимального) времени разработки версии ПО (модификации ПО) с длительным, итерационным процессом разработки на основе аппарата «Network calculus». Метод особенно актуален для больших промышленных систем с высоким экономическим риском в случае простоя и объектов повышенного риска эксплуатации.

Исследована применимость метода для расчета максимального времени модификации программного обеспечения АСУТП АЭС на этапах пуска-наладки и эксплуатации для различных способов модификации. Полученные численные значения подтвердились фактическими работами по модификации ПО, время которой, как правило, не превышало 0,7 от этих значений.

ЛИТЕРАТУРА

1. Ройс У. Управление проектами по созданию программного обеспечения. — М.: ЛОРИ, 2002. — 424 с.
2. Голубева Т.А. Структурно-логическая модель прогнозирования времени выполнения проекта разработки програм-

многo обеспечения // Научные труды Винницкого национального техн. ун-та. — 2008. — № 3.

3. Jorgensen M. A review of studies on expert estimation of software development effort // Journal of Systems and Software. — Feb. 2004. — Vol. 70, iss. 1–2. — P. 37–60.
4. Голубева Т.А., Дубовая Ю.В., Шелест В.С. Нечеткое прогнозирование времени разработки программного обеспечения // Вестник Винницкого политехн. ин-та. — 2011. — № 3. — С. 158–161.
5. Idris A., Amzal F., Abran A. Analogy-based software development effort estimation: A systematic mapping and review // Information and Software Technology. — Feb. 2015. — Vol. 58. — P. 206–230.
6. Royce W. Managing the development of large software systems // Proc. of IEEE WESCON 26. — August 1970. — P. 328–338.
7. Клейнрок Л. Теория массового обслуживания. Пер. с англ. — М.: Машиностроение, 1979. — 432 с.
8. Абу-Абед Ф.Н., Глодева Е.А., Допира Р.В. Многофазные модели эксплуатации сложных технических систем // Проблемы информатики в образовании, управлении, экономике и технике, Пенза, 13–14 ноября 2014 г.: Сб. статей XIV междунар. науч.-техн. конф., Пенза, 2014. — С. 15–21.
9. Опыт проектирования и внедрения системы верхнего блочного уровня АСУТП АЭС / М.Е. Бывайков, Е.Ф. Жарко, Н.Э. Менгазетдинов и др. // Автоматика и телемеханика. — 2006. — № 5. — С. 65–79.
10. Le Boudec J.-Y., Thiran P. Network Calculus: A Theory of Deterministic Queuing Systems for the Internet/Online Version of the Book Springer Verlag. — LNCS 2050. Version April 26, 2012.
11. Vantanski N., et al. Compensating the transmission delay in networked control systems // 14th Nordic process control workshop, NPCW07, Espoo, Finland, 2007.
12. Hwangnam Kim Hou, J.C. Network calculus based simulation for TCP congestion control: theorems, implementation and evaluation // INFOCOM 2004. Twenty-third Annual Joint Conf. of the IEEE Computer and Communications Societies. Publication Date: 7–11 March, 2004. — Vol. 4. — P. 2844–2855.
13. Масолкин С.И., Промыслов В.Г. Расчет некоторых параметров промышленной вычислительной сети объектов повышенного риска эксплуатации на примере АСУТП АЭС // Проблемы управления. — 2010. — № 1. — С. 47–52.
14. Louis Baccelli et al. Synchronization and Linearity An Algebra for Discrete Event Systems (Wiley Series in Probability and Statistics) — N.-Y.: John Wiley & Sons, 1992. — 514 p.
15. Turner J. New Directions in Communications (or which way to the information age?) // IEEE Communications Magazine. — 1986. — Vol. 24, N 10. — P. 8–15.
16. Sidi M., Liu, W.-Z., Cidon, I., Gopal I. Congestion control through input rate regulation // IEEE Trans. on Communications. — 1993. — Vol. 41, iss. 3. — P. 471–477.
17. Cruz R.L. A Calculus for Network Delay. Part I: Network Elements in Isolation // IEEE Trans. on Information Theory. — Jan. 1991. — Vol. 37. — P. 114–131.
18. Cruz R.L. A calculus for network delay. Part II: Network analysis Information Theory // IEEE Trans. on Information Theory. — Jan. 1991. — Vol. 37. — P. 132–141.
19. Mohd R., Nazir M. Software Reliability Growth Models: Overview and Applications // Journal of Emerging Trends in Computing and Information Sciences. — Sep. 2012. — Vol. 3, N 9. — P. 1309–1320.

Статья представлена к публикации членом редколлегии В.М. Вишневым.

Байбулатов Артур Арсенович — науч. сотрудник,
✉ bajbulatov@mail.ru,

Институт проблем управления им. В.А. Трапезникова РАН,
г. Москва.