

*АВТОМАТИЧЕСКОЕ УПРАВЛЕНИЕ
И ИДЕНТИФИКАЦИЯ*

УДК 621.396

**О ВОЗМОЖНОСТЯХ ПРИМЕНЕНИЯ МЕТОДОВ
И МОДЕЛЕЙ ТЕОРИИ РАСПИСАНИЙ
ДЛЯ ОПТИМИЗАЦИИ РАБОТЫ ВЕБ-СТУДИИ***

Т.В. АВДЕЕНКО¹, Р.В. ПЕТРОВ²

¹ 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, профессор, доктор технических наук. E-mail: tavdeenko@mail.ru

² 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет. E-mail: romanpetrov78@gmail.com

В статье рассматриваются вопросы применения теории расписаний для управления проектами в веб-разработке. Дается обзор литературы. Рассматриваются возможности по применению существующих методов и алгоритмов для построения расписания работы специалистов. Описывается специфика работы веб-студии, способы и форматы решаемых веб-студией задач. Производится формализация задачи для дальнейших исследований. При этом выделяются особенности работы веб-студии: невозможность точного прогнозирования объема задач и доступных ресурсов на длительный срок, потребность в динамическом перестроении расписания в режиме онлайн при поступлении задачи с необходимостью немедленного реагирования. Выделяются и активно применяются в выборе подходящих методов и алгоритмов следующие особенности работы веб-студии по сравнению с классическими задачами теории расписаний. Во-первых, это возможность разбиения проекта на такие задачи, которые могут быть назначены на один ресурс (выполнены одним сотрудником), и ни одна задача не требует назначения на несколько ресурсов. Во-вторых, важной особенностью расписания для веб-студии является уменьшение точности и глубины проработки расписания в зависимости от периода, на которое оно строится. Поэтому предлагается построение нескольких видов расписаний: долгосрочного и краткосрочного. Основные задачи долгосрочного расписания – видеть общую загрузку ресурсов компании в перспективе нескольких месяцев и реагировать на планируемый уровень загрузки путем планирования продаж. Основная задача краткосрочного расписания – непосредственное назначение задач на исполнителей. Промежуточный вариант расписания (от недели до месяца) позволяет спланировать загрузку сотрудников задачами на краткосрочный период и предоставить заказчику статус его проектов и задач.

Ключевые слова: теория расписаний, веб-разработка, управление проектами

DOI: 10.17212/2307-6879-2016-2-7-20

*Статья получена 7 марта 2016 г.

ВВЕДЕНИЕ

Управление проектами представляет собой развитую методологию, включающую совокупность моделей, методов и программных продуктов, применяемых при разработке и реализации проектов различных классов и типов. Важной составляющей данной методологии является управление временными характеристиками проекта. Одним из инструментов управления временными характеристиками проекта являются методы теории расписаний.

В зависимости от предметной области, в которой выполняется проект, могут использоваться различные методы теории расписаний. Такой отраслью является разработка веб-проектов. Проекты в сфере веб-разработки обладают рядом особенностей по сравнению с общей постановкой задач теории расписаний. Часть таких особенностей связана с тем, что большинство веб-студий в России относятся к малому и микробизнесу: по данным исследования [1], в 2013 году 65,5 % веб-студий имеет штат в пределах от 4 до 15 человек, компаний со штатом более 50 сотрудников на рынке не более 3 %, а 12 % – это компании со штатом 1–3 человека, то есть относящиеся к категории «самозанятость». Такая малая численность компании не позволяет иметь в штате выделенных сотрудников, применяющих современные математические методы теории расписаний.

Вместе с тем специфика веб-разработки, когда проект оценивается в человеко-часах, вынуждают веб-студии использовать инструменты планирования занятости сотрудников, постановки и контроля задач и т. д. Согласно исследованию [2], в среднем веб-студия использует три программных инструмента для управления задачами и проектами. Кроме того, активно применяются элементы теории ограничений [3], гибкие методологии разработки [4] и др.

Таким образом, можно предположить, что при наличии адекватного инструмента для автоматизированного составления расписания веб-студии будут применять методы теории расписаний. В данной статье выполняются описание и формализация задач при управлении проектами в веб-разработке.

1. ПОСТАНОВКА ЗАДАЧИ

Веб-разработкой традиционно называют всё, что относится к разработке, поддержке и модернизации сайтов и веб-приложений. Компании, разрабатывающие веб-проекты, традиционно называются веб-студиями. В число сотрудников веб-студии входят программисты, дизайнеры, тестировщики, менеджеры проекта и другой технический персонал. Количество веб-студий в России и СНГ оценивается примерно в 10 000 компаний [5]. Также веб-разработкой могут заниматься внутренние отделы в компаниях – владельцах

веб-проекта. По структуре персонала и решаемым задачам такие отделы сходны с веб-студиями.

Как правило, при оценке длительности проекта или работы используются человеко-часы. При этом большой проект декомпозируется на небольшие работы, доступные для выполнения отдельным специалистом. Помимо проектов, для многих веб-студий актуально выполнение работ в режиме «техподдержки», когда работы являются сравнительно небольшими по затратам времени и поступают на выполнение параллельно выполнению больших проектов.

Таким образом, можно предположить, что для веб-студии будет актуально применение методов теории расписаний для составления календарного плана выполнения работ и оптимизации данного плана. Для этого необходимо сопоставить термины и ресурсы, использующиеся в веб-студии, с терминами и объектами теории расписаний и определить постановку задачи для дальнейшего исследования.

2. ТЕОРИЯ

Рассмотрим типичный проект по созданию веб-сайта. При создании сайта чаще всего выделяют следующие этапы.

1. Сбор требований.
2. Разработка прототипов ключевых страниц.
3. Разработка технического задания.
4. Разработка дизайна сайта.
5. Верстка дизайна сайта.
6. Программирование и настройка сайта.
7. Тестирование.
8. Запуск проекта в эксплуатацию.
9. Гарантийная поддержка.

Схематично данные этапы можно представить на рисунке (сроки выполнения работы указаны условные).

	07 Окт '13	14 Окт '13	21 Окт '13	28 Окт '13	04 Ноя '13	11 Ноя '13	18 Ноя '13	25 Ноя '13	02 Дек '13	09 Дек '13	16 Дек '13	23 Дек '13	30 Дек '13	06 Янв '14	13 Янв '14	20 Янв '14	27 Янв '14	03 Фев '14	10 Фев '14	17 Фев '14		
Начало Вт 01.10.13	Сбор требований Вт 01.10.13 - Чт 10.10.13		Разработка прототипов ключевых страниц Пт 11.10.13 - Чт 31.10.13		Разработка ТЗ Пт 01.11.13 - Чт 14.11.13		Дизайн сайта Пт 15.11.13 - Чт 12.12.13			Верстка дизайна Пт 13.12.13 - Чт 02.01.14			Программирование, настройка сайта Пт 03.01.14 - Чт 30.01.14			Тести- рова- ние Пт 06.02.14 - Вт 31.01.14 - Вт 18.02.14		Напол- нение Чт 12.02.14 - Вт 11.02.14		Запуск Ср 18.02.14		Оконча- ние Вт 18.02.14
	4		4		4		4			4			4		4		4		4		4	

Этапы создания сайта

Построение расписания для такого проекта – это задача построения последовательности выполнения работ проекта с учетом отношений предшествования и ограничения на ресурсы (Resource – Constrained Project Scheduling Problem, RCPSP) [6]. Выделим основные особенности ведения проектов в веб-разработке.

1. На разных этапах проекта задействуются различные специалисты, в общем случае не связанные между собой.

2. Формально на рисунке каждый следующий этап начинается сразу после окончания предыдущего. Реально начало следующего этапа часто зависит от согласования результатов и подписания актов по предыдущему этапу, поступления предоплаты и наличия свободных ресурсов у исполнителя.

3. Перед началом проекта выполняется этап продажи проекта и предварительной аналитики (не показаны на рисунке), на который составляется ориентировочная смета с трудозатратами (в нормо-часах) и временными затратами (в днях) проекта. Иногда на этапе продажи составляется предварительный календарный план-график проекта, на который стороны опираются в оценках завершения проекта. При этом традиционно на этапе продажи принято умалчивать или занижать риски задержки выполнения проекта. Часто используется методика, когда в предварительный календарный план заносится чистое время работы (нормодни), без учетов дней согласования, задержек предоставления материалов, отпусков, болезней сотрудников и т. д. По нашему опыту, увеличение продолжительности проекта по разработке сайта в среднем составляет 1,5–2 раза от декларируемого «чистого времени работы».

4. В приведенном на рисунке временном графике этапы идут «без остановки». Однако, в реальности ряд этапов проекта вынужден приостанавливаться на получение обратной связи от заказчика и затем возобновляться. Это необходимо учитывать при составлении расписания.

5. Средний проект по разработке индивидуального веб-проекта занимает от 6 до 12 месяцев.

6. Если за 1–2 месяца известно, что у веб-студии будет недостаток ресурсов для выполнения запланированного объема работ, то за указанное время можно найти либо субподрядчика, либо сотрудника в штат.

7. В среднем в отрасли сотрудник меняет работу каждые 2 года. Поэтому, если в компании более 10 сотрудников, занятых в производстве, то планирование работы конкретного работника на срок более трех месяцев затруднительно (возможно, к этому времени он уволится).

8. Для многих веб-студий велика вероятность появления незапланированных заранее задач, которые нужно внести в ближайшее расписание. Такими задачами, например, являются задачи по исправлению аварийных ситуаций (когда сайт клиента перестал работать).

Таким образом, в силу специфики для веб-студии актуально как составление долгосрочного расписания с достаточно примерной загрузкой ресурсов (период 1-6 месяцев, чем дальше от текущей даты, тем более неточно расписание), так и краткосрочного (на ближайшую неделю и месяц) с более точным распределением нагрузки на ресурсы.

3. ПРИМЕНЯЕМЫЕ МЕТОДЫ И АЛГОРИТМЫ

Построение долгосрочного расписания

Для построения долгосрочного расписания воспользуемся постановкой задачи RCPSP [6].

Дано множество требований $N = \{1, \dots, n\}$ и K возобновляемых ресурсов $k = \{1, \dots, K\}$. В каждый момент времени t доступно Q_k единиц ресурса k . Заданы продолжительности обслуживания $p_i \geq 0$ для каждого требования $i = \{1, \dots, n\}$. Во время обслуживания требования i требуется $q_{ik} \leq Q_k$ единиц ресурса. После завершения обслуживания требования освобожденные ресурсы в полном объеме могут быть мгновенно назначены на обслуживание других требований.

Между некоторыми парами требований заданы ограничения предшествования: $i \rightarrow j$ означает, что обслуживание требования j начинается не раньше окончания обслуживания требования i . Обслуживание требований начинается в момент времени $t = 0$. Прерывания при обслуживании требований запрещены.

Необходимо определить моменты времени начала обслуживания требований $S_i, i = \{1, \dots, n\}$ так, чтобы минимизировать время выполнения всего проекта, т. е. минимизировать значение

$$C_{\max} = \max_{i=1, \dots, n} \{C_i\},$$

где $C_i = S_i + p_i$. При этом должны быть соблюдены следующие ограничения:

1) в каждый момент времени $t \in [0, C_{\max})$ должно выполняться условие $\sum_{i=1}^n q_{ik} \varphi_i(t) \leq Q_k$, где $\varphi_i(t) = 1$, если требование i обслуживается в момент времени t и $\varphi_i(t) = 0$ – в противном случае. То есть требования в процессе своего обслуживания должны быть полностью обеспечены ресурсами;

2) не нарушаются отношения предшествования между требованиями, т. е. $S_i + p_i \leq S_j$, если $i \rightarrow j$ для $i, j \in N$.

Как отмечается в [6], «на практике задача RCPSP обладает рядом отличий от математической модели». Данная постановка применима для построения долгосрочного расписания веб-студии при введении следующих допущений.

1. Для учета плановых остановок и возобновления проекта после получения обратной связи от заказчика можно использовать следующий прием: если из прошлого опыта известно, что продолжительность выполнения определенного этапа в M раз больше, чем запланированная по нормо-дням, при этом количество нормо-часов на выполнение этапа не увеличивается (т. е. имеются остановки в выполнении этапа), то можно считать, что вместо одной единицы ресурса K_i с производительностью p_i есть M ресурсов K_{ij} , $j = 1, \dots, M$ производительностью p_{ij} , где $\sum_{j=1}^M p_{ij} = p_i$, и у них отсутствуют остановки.

2. Для учета задержки выполнения следующего этапа после окончания предыдущего также можно учитывать эмпирический опыт и ввести дополнительные этапы «подготовка к этапу дизайна», «подготовка к этапу программирования» и т. д. У данных этапов задается либо фиксированное время выполнения, определенное эмпирически, либо данный этап относится к типу «гамак» [6].

3. Несмотря на то что проект утвержден, на практике этапы проекта могут сдвигаться во времени или совсем отменяться (если заказчик решил остановить проект). Это также возможно сделать в рассматриваемой постановке, увеличив количество доступных долгосрочных ресурсов на определенную величину.

Таким образом, в долгосрочном расписании необходимо видеть, насколько веб-студия обеспечена ресурсами и работой. Точное распределение задач по исполнителям не требуется.

Построение краткосрочного расписания

Рассмотрим построение краткосрочного расписания. К моменту реального начала соответствующего этапа проекта большая часть работ выполнена (например, «разработка дизайна сайта» уже детализирована до более мелких задач, например «Разработка дизайна главной страницы», «разработка дизайна страницы списка новостей», разработка «дизайна страницы новости»). По каждой из этих задач есть корректные плановые трудозатраты и плановые сроки выполнения и в общем случае выполняется требование «прерывания при обслуживании задачи запрещены».

Таким образом, и при построении краткосрочного расписания постановка задачи RCPSP в общем случае является корректной. Рассмотрим ее конкретизацию применительно к нашей ситуации.

1. В общей постановке задач RCPSP допустимо выполнение одной задачи несколькими ресурсами (например, перевозка стройматериалов может быть выполнена тремя, пятью, семью грузовиками и т. д.). В веб-разработке, даже если задача поручается двум сотрудникам, каждому из них поручается индивидуальная задача, т. е. в этом случае одна задача разбивается на две. Таким образом, задачи в проекте можно рассматривать как назначаемые только на один ресурс.

2. Количество доступных ресурсов является функцией от времени: сотрудники болеют, уходят в отпуск и т. д.

3. Продолжительность выполнения задачи зависит от ресурса (сотрудника), который ее будет выполнять. Чем выше квалификация сотрудника, тем, как правило, быстрее будет сделана задача.

4. У сотрудников тоже есть время на «переналадку», т. е. вникание в другую задачу. Как правило, это не менее 15 минут.

5. Задачи поступают не в начальный момент времени, а по мере возникновения.

6. Во многих проектах можно сдавать («запускать») проект «по частям», тем самым сдвигая временные границы завершения всего проекта на более поздний срок.

7. Даже если расписание составлено, могут неожиданно появиться новые задачи, отсутствующие ранее (например, восстановление работы сайта клиента). Их нужно вносить в расписание на ближайшее время.

Отметим, что в веб-студии чаще всего назначение задач выполняет специальный сотрудник – диспетчер. Возможен вариант, когда эту роль делят между собой несколько сотрудников.

В работах [6] и [7] приводится ряд алгоритмов, которыми можно воспользоваться для составления краткосрочного расписания и диспетчеризации задач.

Например, в работе [6] приводится алгоритм диспетчеризации ListScheduling (LS), который позволяет строить активные расписания. При этом авторы признают, что в зависимости от выбора начальной задачи мы можем либо получить оптимальное расписание, либо остаться с активным расписанием. Далее там же предлагается ряд правил выбора начальной задачи, в зависимости от которых мы получим различное решение. При этом нет гарантии, что все задачи завершатся в их директивные сроки завершения.

Эти ограничения можно обойти с помощью алгоритма, представленного в исследовании [7]. Указанный алгоритм производит построение расписания для проекта «с конца», т. е. с конечных задач, и привязывается к директивным срокам выполнения. При этом в алгоритме присутствует обработка исключи-

тельной ситуации, когда не удастся построить расписание, удовлетворяющее всем директивным срокам окончания задач.

Рассмотрим указанные алгоритмы.

Алгоритм ListScheduling (LS) [6]

1. Пусть EL – список всех требований без предшественников. Полагаем $Q_k(\tau) = Q_k \forall \tau, k = 1, \dots, K$.
2. **while** $EL \neq 0$ **do**.
3. Выберем требование $j \in EL$.
4. $t := \max_{i:(i,j) \in A} \{S_i + p_i\}$. Если для требования j предшественники не определены, тогда $t := 0$.
5. **while** существует такой ресурс k , что для $q_{jk} > Q_k(\tau)$ для некоторого $\tau \in [t, t + p_j)$ **do**.
6. Вычислим такое минимальное значение $t_k > t$, что требование j может быть обслужено на интервале $[t_k, t_k + p_j)$, если рассматривается только ресурс k .
7. $t := t_k$.
8. **endwhile**.
9. Назначим выполнение требования j на интервал $[S_j, C_j) = [t, t + p_j)$.
10. Резервируем ресурсы под выполнение требования j :
 $Q_k(\tau) = Q_k(\tau) - q_{jk}, k = 1, \dots, K, \tau \in [t, t + p_j)$.
11. $EL = EL \setminus \{j\}$.
12. Добавляем в EL последователей требования j , для которых все предшественники поставлены в расписание.
13. **endwhile**.

Алгоритм построения расписаний «с конца» [7]

Для применения данного алгоритма сначала необходимо построить упорядоченный список последовательностей работ. Первые позиции в этой последовательности занимает подпоследовательность $s1$, готовая к выполнению в момент $d = 0$ и имеющая наивысший приоритет из всех возможных допустимых подпоследовательностей, организованных на множестве ресурсов J , готовых к выполнению в момент $d = 0$. За подпоследовательностью $s1$ следу-

ет подпоследовательность s_2 , построенная по аналогичным правилам, готовая к выполнению после выполнения s_1 и имеющая высший приоритет из допустимых подпоследовательностей, организованных на множестве ресурсов $J \setminus s_1$.

Алгоритм

1. Инициализация и очистка списков. Создание пустого списка ресурсов, без назначенных работ.

2. Выбираем все конечные работы, заносим их в список неназначенных работ. Величина «позднее время окончания» для конечных работ равна директивному сроку задания.

3. Если список неназначенных работ пуст, переход на п. 7, иначе – на п. 4.

4. Найти в списке неназначенных работ элемент с минимальным номером подпоследовательности и максимальным значением времени запуска (позднее время окончания, определяемое минимальным временем запуска преемников, минус длительность). Запомнить номер соответствующей ему вершины для дальнейшего использования.

5. Назначение найденной работы, т. е. перенос ее из списка неназначенных работ в список назначенных, следующим образом. Для каждого однотипного ресурса ищем по списку назначенных работ место для вставки работы, просматривая в «реальном» времени ресурса в порядке с конца в начало в поиске резерва, оканчивающегося не позже, чем позднее время окончания назначаемой работы, и длиной не менее его длительности.

Если резерв найден, переход на п. 6. Если требуемый резерв не найден во всех однотипных ресурсах, то пользователю выдается сообщение о том, что выполнение данной работы не укладывается в директивный срок задания. Пользователь должен принять соответствующие меры, переход на п. 1.

6. Выбор всех предшественников сохраненной вершины и добавление их в список неназначенных работ по общим правилам, определив позднее время окончания как разность времени окончания сохраненной вершины и ее длительности. Переход на п. 3.

7. Сформировать список назначенных работ с информацией о временах запуска и окончания (план выполнения работ).

Отметим, что помимо приведенных алгоритмов, существуют и другие алгоритмы. Например, в работах [8, 9] рассматриваются вопросы по разработке эффективных вычислительных методов решения задач теории расписаний для случая параллельных задач. Такие задачи являются NP-сложными задачами. Также одним из эффективных методов решения задач теории расписаний является мультиагентный подход [10]. Все эти методы могут применяться при построении расписания на тот или иной период.

4. ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

Для построения расписания в веб-разработке необходимо использовать два вида расписаний. Долгосрочное расписание – для планирования общей загрузки ресурсов компании по проектам на период 1-6 месяцев. Краткосрочное – для построения конкретного детального расписания сотрудников на период от одного дня до месяца.

Для этого можно использовать описанные в работах [6, 7] алгоритмы, вбирая начальные задачи на основе текущих приоритетов компании, а в перспективе – применять методы и подходы [8–11].

При этом необходимо учитывать ряд особенностей веб-разработки, которые позволяют упростить используемые алгоритмы. Ключевой особенностью является возможность разбиения проекта на такие задачи, которые могут быть назначены на один ресурс (выполнены одним сотрудником), и ни одна задача не требует назначения на несколько ресурсов.

Еще одной важной особенностью расписания для веб-студии является уменьшение точности и глубины проработки расписания в зависимости от периода, на которое оно строится.

Так, расписание на 1–5 дней строится с разбивкой на конкретные задачи (каждая от одного до восьми часов), конкретных сотрудников и конкретные сроки на каждую задачу. Данное расписание должно быть оптимальным или приближенным к нему. В зависимости от наличия или отсутствия задач с директивным сроком завершения могут быть выбраны разные подходы к построению расписания. Расписание на 5-30 дней строится уже на более крупные блоки задач (каждая на 4-20 часов), с уменьшенным уровнем детализации по времени и сотрудникам и допустимой неоптимальностью расписания.

Расписание на 30–90 дней строится на еще более крупные блоки задач, уже возможно без детализации по конкретным сотрудникам и с возможностью нарушения ряда условий задачи RCPSP (например, обеспеченности ресурсами). Расписание на 90–180 дней строится очень крупными блоками.

При этом используемый алгоритм должен предупреждать о потенциальных проблемах в расписании на будущие периоды для того, чтобы руководители проектов и руководители компании могли принять меры по увеличению доступных ресурсов (в случае превышения объема планируемых задач над доступными ресурсами) или по увеличению продаж (в случае недостаточного объема планируемых задач для загрузки имеющихся ресурсов).

Также актуальны «вставка» в расписание новых работ, отсутствующих ранее (алгоритм должен уметь это делать), и визуализация текущих назначенных задач на исполнителя и статусов этих задач.

ЗАКЛЮЧЕНИЕ

Несмотря на широкое использование средств автоматизации и учета рабочего времени по задачам, в веб-студиях актуально применение методов теории расписаний для оптимизации графика выполнения проектов. При этом в профильной литературе достаточно много публикаций на тему управления проектами в крупных компаниях и недостаточно внимания уделено работе с малым бизнесом. Поэтому адаптация и развитие методов теории расписаний для рынка веб-разработки имеет широкие перспективы для исследований.

СПИСОК ЛИТЕРАТУРЫ

1. Рынок веб-разработки в Рунете: тренды и ожидания 2013 года [Электронный ресурс] // Рейтинг Рунета: web-сайт. – М., 2013. – URL: <http://www.ratingruneta.ru/research/webanalytics2012/> (дата обращения: 20.06.2016).
2. Рейтинг сервисов для автоматизации таск-трекинга 2016 [Электронный ресурс] // Tagline: web-сайт. – 26 апреля 2016. – URL: <http://tagline.ru/task-bug-tracking-services-rating> (дата обращения: 20.06.2016).
3. *Голдратт Э.М.* Критическая цепь. – Минск: Попурри, 2013. – 240 с.
4. Гибкая методология разработки [Электронный ресурс] // Википедия: web-сайт. – URL: https://ru.wikipedia.org/wiki/Гибкая_методология_разработки (дата обращения: 20.06.2016).
5. CMSMagazine: аналитический портал рынка веб-разработок [Электронный ресурс]. – URL: <http://www.cmsmagazine.ru/> (дата обращения: 20.06.2016).
6. *Лазарев А.А., Гафаров Е.Р.* Теория расписаний. Задачи и алгоритмы: учебное пособие. – М.: Изд-во МГУ, 2011. – 222 с.
7. *Павлов А.А., Чернов С.К., Мисюра О.Б.* Модели и алгоритмы теории расписаний в задачах планирования и управления проектами // Труды Одесского политехнического университета. – 2006. – Вып. 1 (25). – С. 150–159.
8. *Mezentsev Yu.A., Avdeenko T.V.* Scheduling and optimization for parallel-serial service systems // Proceedings of the 8th International Forum on Strategic Technology (IFOST 2013), 28 June –1 July, Mongolian University of Science and Technology. – Ulaanbaator, Mongolia: IEEE, 2013. – Vol. 1. – P. 271–275.
9. *Avdeenko T.V., Mezentsev Yu.A.* Efficient approaches to scheduling for unrelated parallel machines with release dates // Proceedings International Conference on Manufacturing Modeling, Management & Control (MIM 2016), 28–30 June 2016. – Troyse, France: UTT, 2016.

10. Авдеенко Т.В., Васильев М.А. Мультиагентный подход с использованием нечеткого моделирования в задачах многокритериального принятия решений // Научный вестник Новосибирского государственного технического университета. – 2010. – № 1. – С. 63–74.

11. Авдеенко Т.В. Проблемы параметрической идентификации в математическом моделировании процессов // Образовательные ресурсы и технологии. – 2014. – № 1 (4). – С. 115–124.

Авдеенко Татьяна Владимировна, доктор технических наук, профессор, заведующая кафедрой экономической информатики Новосибирского государственного технического университета. Является автором около 140 научных работ, включая монографию и 6 учебных пособий. Область научных интересов – моделирование и идентификация, методы поддержки принятия решений. E-mail: tavdeenko@mail.ru

Петров Роман Владимирович, с 2006 года руководит веб-студией «Ай ТиКонстракт», которая входит в число ведущих веб-студий России (по рейтингам «Тэглайн» и «Рейтинг Рунета»). Автор учебных курсов по веб-разработке для студентов НГТУ. E-mail: romanpetrov78@gmail.com

On the possibility of applying methods and models of scheduling theory to optimization of working a web-studio *

T.V. Avdeenko¹, R.V. Petrov²

¹ *Novosibirsk State Technical University, 20 K. Marx Prospekt, Novosibirsk, 630073, Russian Federation, professor, doctor of technical science. E-mail: tavdeenko@mail.ru*

² *Novosibirsk State Technical University, 20 K. Marx Prospekt, Novosibirsk, 630073, Russian Federation. E-mail: romanpetrov78@gmail.com*

The article deals with the application of scheduling theory for project management in web development. A review of the literature is given. The possibilities for application of the existing methods and algorithms for the construction of schedules are considered. We describe the specifics of working, the methods and formats of the problems solved by web-studio. Formalization of problems for further research is made. We also point out the features of the web-studio: the inability to accurately predict the volume of work and available resources for the long term, the need to build a dynamic schedule on-line when entering current job with the need to respond immediately. We separate the following features of the web-studio in comparison with classical scheduling problems and actively use them in the selection of appropriate methods and algorithms. Firstly, it is an opportunity to divide the project into subproblems

* Received 07 March 2016.

which can be assigned to one resource (performed by one staff member), and no problem requires assigning to several resources. Second, an important feature of the schedule for the web-studio is a decrease in the accuracy and depth of schedules depending on the period for which it is built. Therefore several types of schedule are proposed: long-term and short-term. The main objective of the long-term schedule is to see the overall workload of resources for the period of a few months and to respond to the planned level by sales planning. The main objective of short-term schedules is direct assignment of tasks to the performers. An intermediate version of the schedule (from a week to a month) allows to plan the loading for the employees in the short term and to provide the customer with the status of its projects and tasks.

Keywords: scheduling theory, project management, web-development

DOI: 10.17212/2307-6879-2016-2-7-20

REFERENCES

1. Rynok veb-razrabotki v Runete: trendy i ozhidaniya 2013 goda [The market of web development in RuNet: trends and expectations in 2013]. *Reiting Runeta* [Rating Runet]. Available at: <http://www.ratingruneta.ru/research/webanalytics2012/> (accessed 20.06.2016)
2. Reitingi servisov dlya avtomatizatsii task-trekinga 2016 [Rating of services to automate task-tracking 2016]. *Tagline*, 26 April, 2016 (accessed 20.06.2016)
3. Goldratt E.M. *Critical chain*. Great Barrington, North River Press, 1997. 246 p. (Russ. ed.: Goldratt E.M. *Kriticheskaya tsep'*. Minsk, Popurri Publ., 2013. 240 p).
4. Gibkaya metodologiya razrabotki [Agile software development]. *Vikipediya* [Wikipedia]. Available at: https://ru.wikipedia.org/wiki/Гибкая_методология_разработки (accessed 20.06.2016)
5. *CMSMagazine: Analytical portal of market of web development*. (In Russian) Available at: <http://www.cmsmagazine.ru/> (accessed 20.06.2016)
6. Lazarev A.A., Gafarov E.R. *Teoriya raspisanii. Zadachi i algoritmy* [Scheduling theory. Tasks and algorithms]. Moscow, MSU Publ., 2011. 222 p.
7. Pavlov A.A., Chernov S.K., Misyura O.B. Modeli i algoritmy teorii raspisanii v zadachakh planirovaniya i upravleniya proektami. *Trudy Odesskogo politekhnicheskogo universiteta – Works of the Odessa polytechnic university*, 2006, iss. 1 (25), pp. 150–159.
8. Mezentsev Yu.A., Avdeenko T.V. Scheduling and optimization for parallel-serial service systems. *Proceedings of the 8th International Forum on Strategic Technologies (IFOST 2013)*, Ulaanbaatar, Mongolia, 28 June – 1 July 2013, vol. 1, pp. 271–275.
9. Avdeenko T.V., Mezentsev Yu.A. Efficient approaches to scheduling for unrelated parallel machines with release dates. *Proceedings International Conference*

on *Manufacturing Modeling, Management & Control (MIM 2016)*, 28–30 June 2016, Troyes, France, UTT, 2016.

10. Avdeenko T.V., Vasil'ev M.A. Mul'tiagentnyi podkhod s ispol'zovaniem nechetkogo modelirovaniya v zadachakh mnogokriterial'nogo prinyatiya reshenii [Multiagent approach with use of fuzzy modeling in the task multicriterion decision making]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2010, no. 1, pp. 63–74.

11. Avdeenko T.V. Problemy parametriceskoi identifikatsii v matematicheskom modelirovanii protsessov [Parametric identification problems in mathematical modeling of processes]. *Obrazovatel'nye resursy i tekhnologii – Educational resources and technology*, 2014, no. 1 (4), pp. 115–124.