

Программный продукт для планирования работы сотрудников IT-компаний

В.В. Попов, email: corybook05@yandex.ru¹
И.В. Попова, email: corybook05@yandex.ru²

¹ ВУНЦ ВВС «ВВА имени проф. Н.Е. Жуковского и Ю.А. Гагарина»
(г. Воронеж)

² Воронежский государственный университет

***Аннотация.** В данной работе рассматривается процесс разработки программного продукта для планирования и анализа работы сотрудников, а также распределения IT-проектов.*

***Ключевые слова:** IT-проект, коэффициент эффективности, область знаний, планирование проектов, SAP.*

Введение

Успех работы IT-компаний во многом зависит от эффективности использования в ней методов оперативного и стратегического планирования. На данный момент в сфере IT-менеджмента данная тема реализована не в полной мере. Однако, именно применение оперативного и стратегического планирования позволяет не только увеличить производительность, а значит, и прибыль, но и проводить анализ выполненной работы, что помогает определять ключевые направления развития IT-компаний в будущем [1, 2].

В работе поставлена и решена задача разработки программного продукта для планирования и анализа работы сотрудников, а также распределения IT-проектов.

Актуальность данной задачи обоснована возможностью существенного уменьшения временных затрат на выполнение IT-проектов и повышение качества анализа работы сотрудников.

1. Проблемы планирования в IT-компаниях

Для учета времени работы программистов используется ряд специализированных систем оперативного планирования. Эти приложения включают следующий функционал [3]:

- добавление часов на проект;
- запись отработанных часов;
- проверка записанных часов;
- создание отчета для клиента;

просмотр статистики отработанного времени по проекту;
внесение запроса на отпуск;
отображение списка сотрудников с ближайшими датами отпуска;
возможность указать даты выходных, праздников, больничных.

Стоит отметить, что во многих системах тайм менеджмента существует возможность анализа эффективности работы программиста: сколько часов он отработал и на каких проектах, а сколько болел или был в отпуске. Регистрация времени является одним из важнейших требований для каждой IT-компания. Именно учет затраченного на работу времени позволяет создавать счета на оплату для клиентов, а также рассчитывать зарплату и премии для программистов.

Примером может служить следующая ситуация: два программиста работают над одним и тем же заданием. Первый выполняет задание за 5 рабочих дней (40 часов), написав код высокого качества. Второй программист написал код более низкого качества, возможно он не такой быстродействующий, но зато он затратил на это задание 3 рабочих дня (24 часа). Тут следует учитывать, что для руководителя на проекте важнее – качественный быстродействующий код или более быстрое время выполнения данного задания. Конечно, при добавлении новых возможностей в уже существующем приложении или написании приложения с нуля важнее первый вариант, но, если проблема возникла в продуктивной системе и необходимо срочно ее исправить, отдается преимущество второму программисту.

Более того, данные системы помогают учитывать отработанное время в случае отсутствия заданий, т.к. сотрудник обязан описать темы, которые он изучал. Учет продуктивного времени и времени, потраченного на обучение и отдых (отпуск, б/с), позволяет грамотно рассчитывать премии.

Другой проблемой для многих растущих IT-компаний является учет проектов. По мере увеличения количества сотрудников, увеличивается и количество проектов, а денежные потоки становится трудно контролировать. Грамотное распределение проектов, ведение ежедневного учета работы сотрудников, учет ежемесячных затрат, становится тяжело осуществлять без специальных программ. Еще более остро встает вопрос о прогнозируемых проектах, а значит о прогнозируемой прибыли и затратах. Здесь особенно важно использовать приложения стратегического планирования. Такие приложения должны обладать функционалом, позволяющим заносить прогнозируемые проекты (учитывая ресурсы, требуемые на выполнение заданий), вычислять рентабельность проекта и прибыль. Другой важной функцией является вычисление в реальном времени стоимости

отработанных часов, оставшийся бюджет на проект и предупреждение о приближении к окончанию бюджета.

Анализ работы программистов IT-компаний показал, что одной из важных и актуальных проблем является объективное определение уровня значимости или эффективности сотрудника. В настоящее время это является необходимым критерием для планирования проектов, а также выплаты зарплаты и возможности выплаты премий. Один из наиболее востребованных способов оценивания уровня работы программиста – это оценка его от уровня «Junior» до «Lead». Уровень программиста по данной шкале используется практически во всех IT-компаниях, что облегчает поиск нужного кандидата и упрощает понимание его знаний. Однако, уровень у двух программистов может быть одинаковый, а пользу компании они могут приносить разную. Таким образом, ограничиться только одной оценкой в каждой категории от «Junior» до «Lead» является не вполне оптимальным. В некоторых компаниях сотрудников оценивают по отзывам коллег, в других вычисляют «коэффициент эффективности» работы программиста. «Коэффициент эффективности» – это коэффициент, показывающий на сколько эффективно работает программист. Данный коэффициент используется для вычисления рабочего времени на проект (продуктивное время), а также учитывается при определении оплаты труда. Рассмотрим пример компании, которая использует методы оценки уровней от «Junior» до «Lead». При таком подходе для каждого уровня или категории программистов назначается определенный фиксированный (статичный) коэффициент значимости. Коэффициент эффективности позволяет соотнести эффективность работы программистов с разным уровнем подготовки. Необходимо отметить, что при данном подходе не учитываются личные качества программистов внутри одной категории.

Другим важным критерием при распределении проектов является область знаний программиста – в какой области сотрудник разбирается лучше всего, а в какой сможет выполнить только самые простые задачи. Данные сведения позволяют менеджеру проектов быстрее распределить проекты, но что более важно, увидеть, в каких областях недостаточно специалистов. Таким образом, при нехватке ресурсов в определенной области, руководитель, подобрав нужные курсы, может отправить программиста на обучение – повышение квалификации.

В настоящее время определены следующие области знаний: Srm, Erp, Workflow, Webdynpro, Fiori, Javascript, Hana, SAP Cloud Platform, RPA.

В каждой компании определяется свой актуальный на данный момент перечень.

Для исследуемой компании штат которой составляет 10 программистов такой перечень представлен в таблице 1.

Таблица 1

Компетентность программистов в областях знаний (в баллах)

	SRM	ERP	Workflow	Webdynpro	Fiori	JavaScript	Hana	SAPCloudPlatf orm
Программист 1	5	4	5	3	1	1	5	1
Программист 2	4	4	3	3	2	2	5	2
Программист 3	2	2	2	1	5	5	4	5
Программист 4	2	1	5	3	1	2	2	2
Программист 5	5	5	5	5	4	4	5	4
Программист 6	2	2	1	1	2	2	1	2
Программист 7	4	3	5	3	2	3	4	3
Программист 8	4	4	3	5	2	3	4	5
Программист 9	1	3	1	4	1	1	5	1
Программист 10	2	3	4	2	3	4	2	1

Таблица показывает знаниями каких программных продуктов обладает определенный состав сотрудников компании и позволяет определить слабые места в некоторых областях знаний.

2. Анализ загрузки сотрудников ИТ-компаний

В процессе анализа существующего бизнес процесса невозможно обойтись без вычисления загрузки сотрудников на данный момент и разницы между спланированной и выполненной работой [3, 4].

Для вычисления этих данных были использованы данные компании из систем управления рабочим временем сотрудников и планирования сотрудников на проект. Выборка данных проводилась в зависимости от уровня программиста и области знаний.

На рисунке 1 представлена процентная загрузка программиста продуктивными заданиями в зависимости от категории. Показано, что самыми востребованными являются программисты уровня «Lead». Их загрузка составляет около 98%. Следующими являются сотрудники с уровнями «Middle» и «Senior». Процент загрузки у первых всего на 0,5% больше. А вот процент загрузки программиста уровня «Junior» меньше примерно на 7% и составляет 78,85%.

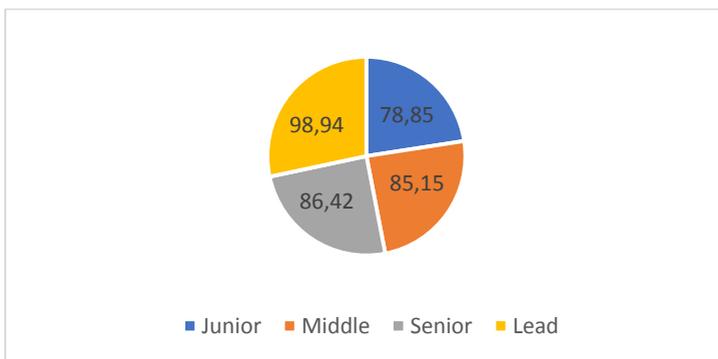


Рис. 1. – Загрузка программистов по категориям, в %

На рисунке 2 представлена загрузка программиста в различных областях знаний. Проанализировав загрузку программистов по областям знаний, можно сделать вывод о том, что наиболее востребованными направлениями являются Srm, Fiori, Hana. Наименее востребованы программисты в области Workflow и RPA.

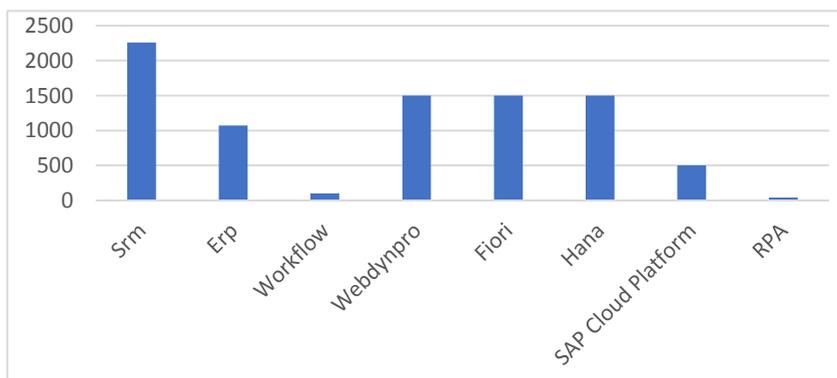


Рис. 2. – Загрузка программистов в зависимости от области знаний, в часах

Представленный анализ показал, что внутренние процессы в организации представляют собой сложную систему, характеризующуюся множеством взаимосвязанных и взаимовлияющих друг на друга свойств. Применение существующих методов оценки эффективности сотрудников работающих в сфере IT-технологий не всегда приводит к удовлетворительным результатам. Обусловлено это

наличием специфических свойств неопределенности и невозможностью учесть все взаимосвязи в некоторой формальной математической модели.

Одним из главных недостатков таких методов является невозможность оценить какой сотрудник работает эффективнее, кому следует выплатить премию, а какой сотрудник работает менее эффективно и не приносит большую пользу для компании. Несмотря на одинаковый уровень профессиональной подготовки один программист может работать более эффективно, чем другой.

3. Анализ существующих решений по планированию

В настоящее время для выполнения планирования существует ряд систем, которые были проанализированы в работе по следующим критериям: простота настройки, простота использования, анализ данных в реальном времени, для больших проектов и для нескольких проектов, расчет оставшегося бюджета в реальном времени, расчет чистой прибыли, расчет рентабельности проекта, интеграция с существующим решением. На основе анализа построена диаграмма, представленная на рисунке 3.

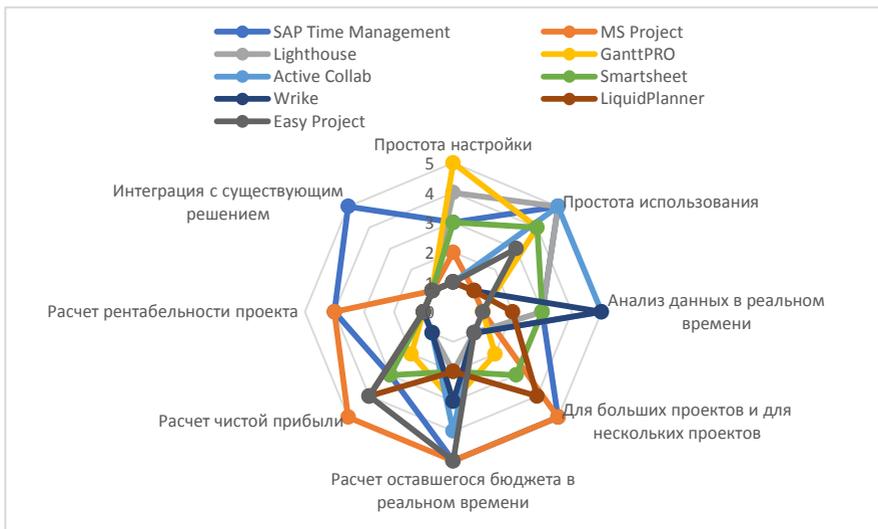


Рис. 3. – Оценка приложений планирования проектов

Полученные результаты показывают, что ни одно из предложенных решений не соответствует всем критериям, которые были выбраны для сравнения. Таким образом, многие компании, понимающие

существенные недостатки представленных выше программ, создают свои программы, помогающие им оценивать и планировать работу сотрудников. Такие программы изначально требуют определенных ресурсов в качестве трудозатрат своих сотрудников, но при этом такие программы включают весь необходимый функционал и учитывают тонкости работы той или иной компании (и ее бизнес процесс).

4. Реализация программного продукта для планирования и анализа работы сотрудников

Создание нового приложения в среде SAP является наиболее оптимальным вариантом. Функционал нового приложения будет отвечать актуальным на сегодняшний день запросам пользователей – руководителей и менеджеров проекта. Также приложение легко интегрируется с уже существующим приложением для учета времени. Еще одним плюсом является сохранение данных о сотрудниках и проектах внутри компании. Большим плюсом является то, что данное решение не будет требовать дополнительных материальных затрат.

Для хранения данных были созданы 5 таблиц (рисунок 4).

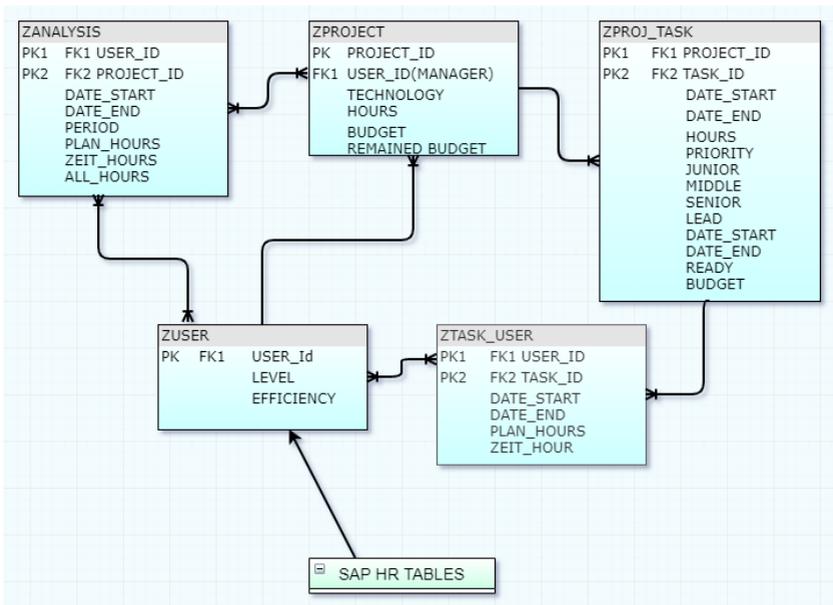


Рис. 4. – Архитектура базы данных

Новое приложение позволяет видеть данные в реальном времени, что позволит менеджеру проекта сразу заметить, когда заканчивается бюджет проекта. Четкий учет и распределение времени по проектам дает возможность оценивать рентабельность проекта и вычисление чистой прибыли.

Архитектура приложения представлена тремя уровнями:

Уровень базы данных – используется система управления реляционной базой данных (PCСУБД SAPR/3).

Сервер приложения – представляет собой классы и методы для работы с базой данных и пользователем.

Сервер представления (клиент) – здесь описан интерфейс приложения для пользователя и логика для отображения UI-элементов на экран.

На этапе построения приложения было принято решение, что система однозначно должна отвечать базовым принципам (инкапсуляция, полиморфизм и наследование). При реализации было определено правило, что каждый метод класса будет отвечать лишь за одно конкретное действие. Архитектура классов представлена на рисунке 5.

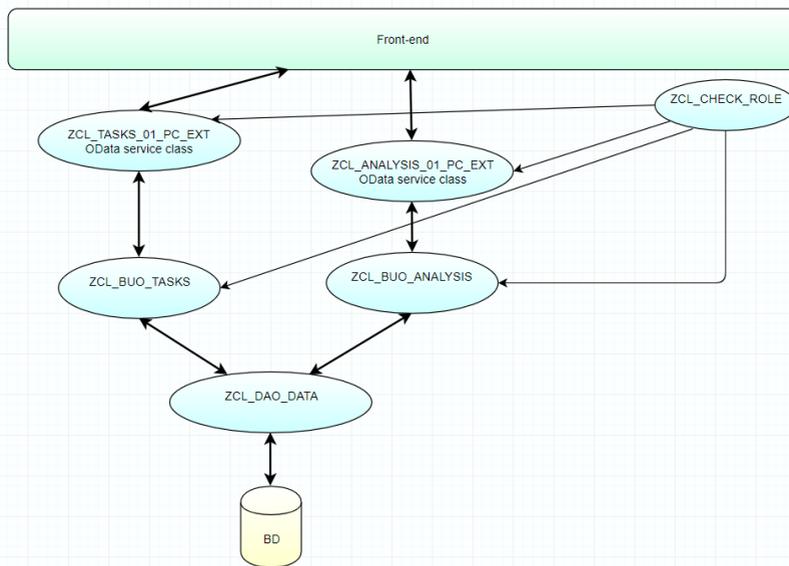


Рис. 5. – Архитектура классов

Для реализации уровня презентации был выбран интерфейс FIORI. Он в свою очередь использует SAP User Interface HTML5. SAPUI5 – это фреймворк для разработки пользовательских интерфейсов на базе кросс-браузерной JavaScript библиотеки [4].

Основные функции, реализованные в приложении: добавить новый проект, запланировать проект, вычисление оставшегося бюджета проекта, анализ в реальном времени – по неделям и месяцам, анализ загруженности каждого пользователя по уровням и по областям знаний, интеграция с приложением учета времени работы сотрудников – данные из него для анализа. Главный экран приложения для пользователя с ролью «Менеджер проекта» представлен на рисунке 6.



Рис. 6. – Главный экран для пользователя «Менеджер проекта»

При нажатии на кнопку «Планирование» открывается новый экран, на котором отображаются сотрудники и соответствующие им проекты. Менеджер проекта может добавлять новые проекты, нажав на кнопку «Добавить проект», а также менеджер может добавить новое задание для проекта.

Заключение

В используемых в настоящее время программных продуктах информация, как правило, актуализируется не достаточно оперативно, а сами системы не интегрированы. Из-за отсутствия интеграции достаточно сложно определить, величину бюджета времени, оставшегося на проект. Отсутствует протокол изменения документа, а сотрудники не вовремя оповещаются об этих изменениях.

Реализация представленного программного продукта планирования и анализа работы сотрудников позволила в некоторой мере устранить описанные выше проблемы. На рисунке 7 показаны результаты загрузки программистов продуктивными заданиями (в процентах) оранжевая линия – до использования разработанного программного продукта, голубая линия – после применения программного продукта.

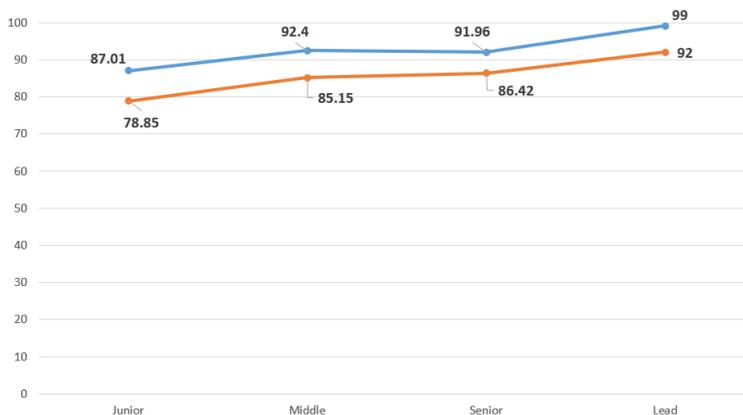


Рис. 7. – Процент продуктивной загрузки программистов

Можно отметить, что продуктивная нагрузка программистов увеличилась в среднем на 7%, что составляет около 9 часов в месяц для одного программиста. Для 10 программистов получаем увеличение продуктивной нагрузки в среднем на 90 часов в месяц.

В работе было определено, что индивидуальная оценка каждого программиста повышает контроль над его загрузкой, результатами выполненной работы и повышает эффективность работы компании в целом. Данный подход позволит увидеть, насколько результативно и быстро справляется с заданиями тот или иной сотрудник, а что более важно, насколько он отдается работе, какие у него достижения, какова его мотивация.

Список литературы

1. Матвеев, М. Г. Методы решения задач управления предприятием в условиях расплывчатой неопределенности / Г. Н. Лебедев, М. Е. Семенов // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. – 2012. – № 1. – С. 102-106.
2. Матвеев, М. Г. Оптимальная производственная стратегия в условиях нечетких параметров функции спроса / Г. Н. Лебедев, М. Е. Семенов // Современная экономика: Проблемы и решения. – 2012. – №6 (30). –С. 214-219.
3. Кречмер, Р. Разработка приложений SAP R/3 на языке АВАР/4 / Р. Кречмер. – СПб.: Питер. 2008. – 334 с.
4. Точенюк, О. Н. Рекомендации по оптимизации программ на языке АВАР / О. Н. Точенюк. – М.: МИР. 2013. – 150 с.