

УДК 004.91 + 004.4'2 + 004.048

## ПРОЕКТИРОВАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ ПРОЕКТАМИ С ЭЛЕМЕНТАМИ АНАЛИТИКИ

*Кисель А.С.*

*Донецкий национальный технический университет*

*Рассматривается вопрос актуальности систем управления разработкой программного обеспечения. Предлагаются идеи расширения функционала с целью повышения эффективности работы.*

### **Введение**

Управление разработкой ПО – это сложный и динамический процесс, в котором принимает участие большое количество людей с разными ролями. Основная задача руководителя проекта – это координация действий сотрудников. В этом и заключается талант руководителя – правильно поставить задачу наиболее подходящему сотруднику. Так как управление проектом – очень изменчивый процесс, очень важно сделать это вовремя.

Потеря гибкости управления однозначно приведет к потерям эффективности производства. Поэтому процесс разработки ПО сегодня просто невозможен без использования системы управления проектами.

**Система управления проектами** представляет собой организационно-технологический комплекс методических, технических, программных и информационных средств, направленный на поддержку и повышение эффективности процессов планирования и управления проектом, в основе которого лежит программное обеспечение календарного планирования.

**Целью** является оценить текущее положение дел в применении информационных технологий в области проектного менеджмента. Попытаться отыскать подходы, которые смогли так или иначе сделать использование подобных инструментов более продуктивным.

**Актуальность** поставленной задачи подтверждается тем, что подобные системы как платные, так и принадлежащие к open source, постоянно совершенствуются и развиваются, а используют их практически все команды разработчиков, независимо от размера и типа разрабатываемого программного обеспечения.

### **1 Потребность в системе управления проектами**

В основе методов планирования и управления проектами лежит предположение, что всю деятельность можно разбить на небольшие, поддающиеся контролю задачи. И тогда всю формальную информацию, используемую менеджером, можно свести к информации о задачах, которые необходимо выполнить для достижения поставленных целей, и ресурсах, которые потребуются для выполнения этих задач. Информационная система управления проектами содержит набор функциональных средств, которые помогают менеджеру спланировать работы, получив временные, ресурсные и стоимостные оценки для комплекса работ, а затем, в процессе выполнения, отслеживать ход работ и корректировать план.

Тщательно составленная план разработке, в виде небольших подзадач, которые по силам выполнить за достигаемый отрезок времени, предоставляет руководству и членам команды возможность принимать приемлемые и обоснованные решения на протяжении всего проектного цикла, наблюдать прогресс разработки, и более-менее точно предполагать дату завершения. Соответствующая организация управления может обеспечить построение процедур анализа качества выполнения работ, которая фокусирует внимание руководства на достижении поставленных целей. В оценке выполненных работ могут принимать участие, как проектный менеджер, так и заказчик.

Информационная модель проекта, разработанная на начальной стадии планирования, подвергается в дальнейшем переработке в процессе его реализации. Таким образом, базовые методики календарного планирования используются на протяжении всего жизненного цикла проекта.

Основные преимущества использования информационной системы для управления проектами включают:

- централизованное хранение информации по графику работ, ресурсам и стоимостям;
- возможности быстрого анализа влияния изменений в графике, ресурсном обеспечении и финансировании на план проекта;
- возможность распределенной поддержки и обновления данных в сетевом режиме;
- возможности автоматизированной генерации отчетов и графических диаграмм, разработки документации по проекту.

## 2 Методологии разработки ПО

Система управления проектами, как правило, реализует одну из методологий разработки ПО. Использует ее термины и реализовывает стандартные для данной методологии подходы.

Большинство крупных систем управления проектами лежат известные методологии разработки. Это объясняется тем, что при создании, в основу трекеров были перенесены те подходы, которые использовались раньше без электронной системы, в бумажном виде.

Методологией разработки программного обеспечения называют набор связанных между собой правил, методов, техник, используемых при разработке ПО. Опубликованные и широко распространённые методологии обычно определяют перечни проектных ролей, типовые задачи разработки с рекомендациями по последовательности их выполнения, перечни промежуточных и окончательных результатов (часто называемых «артефактами»), а также основные правила принятия решений в процессе разработки.

Методологии разработки ПО часто условно разделяют на «тяжёлые» (индустриальные) и «гибкие» (Agile). Рассмотрим подробнее основы гибких методологий.

Манифест Agile (Agile Manifesto), разработанный лидерами Agile-движения, декларирует главные принципы, лежащие в основе гибкого подхода к разработке программного обеспечения.

Манифест Agile заявляет, что для приверженцев гибкой разработки важнее:

- люди и их взаимодействие, чем процессы и средства;
- работающее ПО, чем исчерпывающая документация;
- сотрудничество с заказчиком, чем обсуждение условий контракт;
- реагирование на изменения, чем следование плану.

Рассмотрим далее самые главные, для задачи разработки системы управления, особенности Agile.

1. Планирование процесса (planning game). Вся команда собирается вместе, принимается коллективное решение о том, какие свойства системы будут реализованы в ближайшей итерации. Набор свойств определяется пользовательскими историями. трудоемкость каждого свойства определяется самими программистами.
2. Тесное взаимодействие с заказчиком (feedback, on-site customer). Заказчик должен быть членом команды (on-site customer). Он пишет пользовательские истории, выбирает истории, которые будут реализованы в конкретной итерации, и отвечает на вопросы, касающиеся бизнеса. Заказчик должен быть экспертом в автоматизируемой предметной области. Необходимо постоянное наличие обратной связи с заказчиком (feed-back).
3. 40-часовая рабочая неделя. Программист не должен работать более 8 часов в день. Необходимость сверхурочной работы (overtime) – это четкий индикатор проблемы на данном конкретном направлении разработки. Поиск причин сверхурочной работы и их скорейшее устранение – одно из основных правил

## 3 Разработка собственной системы управления

Проблема любой системы управления на практике то, что разработчики часто не своевременно отмечают результаты своей работы в системе, откладывая это на потом. Учитывая такой подход, говорить дальше о какой-либо аналитике со стороны системы не приходится. Поэтому первым делом следует решить данную проблему неактуальности информации.

Решить вышеописанную проблему предлагается следующим образом. В плане оплаты труда разработчиков очень популярна почасовая оплата. Интегрировав трекер для внесения отработанного времени с указанием задачи, над которой велась работа, можно получить ряд преимуществ:

- сотрудники будут лично заинтересованы в регулярном заполнении табеля. Система, в свою очередь, будет регулярно свежую информацию;
- из табеля времени можно извлечь ряд информации о проекте, часть из которой может пригодиться менеджеру.

Разрабатываемая система должна поддерживать работу на всех этапах разработки. Разделим управление проектом на 3 этапа.

4. Планирование. На основе полученных, на предыдущем этапе (на предыдущей итерации) результатов, менеджером проекта производится оптимальное распределение задач между сотрудниками. Система должна предложить менеджеру оптимальную балансировку, учитывая всю доступную ей информацию о сотрудниках. Распределение должно учитывать время работы сотрудника, его показатели, и уровень. Разумеется, выбор остается за менеджером. Система также должна обучаться на своих ошибках, запоминать, когда предложенные ею варианты были отклонены.
5. Выполнение работы. Сотрудники приступают к выполнению задач, попутно обновляя статусы своих задач. Допускается изменение рабочего плана в процессе работы, снятие или же замена задач другими.
6. Анализ. Производится контроль результатов, сбор статистики. Статистика должна показать отклонение от запланированного графика, граничные показатели. Осуществляется подсчет гонораров сотрудников, на основе их гонорара в час. Также, на основании этого анализа система на следующей итерации должна принимать участие в планировании новых задач. Это может выражаться в подсказках, рассмотренных ниже.

#### 4 Дополнительные функции системы

Сложно без испытания на практике заявлять о пользе дополнительных функций системы. Возможно, они в некоторых ситуациях окажутся неэффективными. Провести подобную проверку на данный момент не представляется возможным, поскольку большинство предлагаемых подходов были разработаны именно для крупной команды разработчиков.

При постановке задач на очередную итерацию, проектному менеджеру может быть представлен ряд информационных подсказок:

- наиболее свободный сотрудник – выбирая кому назначить новое задание система может предложить сотрудника, за которым закреплено меньше всего задач;
- наиболее быстрый сотрудник – выбирать того, кто быстрее всего закрывает задачи;
- выбрать сотрудников с наибольшими отклонениями от нормы, недостаточной/ переработкой отведенных на проекте часов.

Все эти советы могут быть сделаны на основе табеля времени работ. Следовательно, эта информация не требует никаких дополнительных затрат.

Одной из особенностей проектируемой системы будет возможность делать автоматический прогноз количества затрачиваемого сотрудником времени на основе его предыдущих оценок.

Если разработчик работает в одной и той же области и оценивает подряд несколько задач, то с большой вероятностью можно сказать что отношение оцененного количества времени с реальными затратами будет равным. Это может быть применено только к схожим задачам в одной предметной области.

Следует учесть, что в практике таких стандартных задач достаточно. Например: авторизация,

аутентификация, секция администратора, oauth авторизация с помощью социальных сетей и пр.

Разумеется, невозможно гарантировать достоверность подобного прогноза, но в сочетании с другими подходами, он может дополнить общую картину состояния задач.

Подобная статистика разницы, а точнее – динамика изменения этой разницы – отличный способ характеризовать сотрудника. Назовем этот параметр точностью оценки. Чем опытнее сотрудник, тем точнее его оценки должны быть. Точность оценки должен быть одним из основных фактором влияющий на рейтинг сотрудника.

Предварительно оговорим, что при проектировании системы предполагается, что все сотрудники являются профессионалами в своей области. Они дают адекватную оценку поставленным задачам. Но компетентность сотрудника не гарантирует стопроцентного совпадения, и этого никак не избежать. Умение точно определять сроки приходит спустя много лет работы в одной и той же сфере, когда периодически выполняются одни и те же действия. Сотрудник может дать только приблизительную оценку по причинам, описанным ниже.

Порой выполнение задание может сводиться к поиску информации. Даже специалист высокого уровня не сможет точно сказать, за какое время он найдет неизвестную ему до этого информацию.

Бывают моменты, когда для завершения задачи зависит от третьих лиц. Например, необходимо получить подтверждение или какую-либо информацию от заказчика или службы поддержки.

У каждого сотрудника есть склонность преувеличивать или преуменьшать сложность задачи, основанная на личностных качествах.

На погрешность в оценке также влияет человеческий фактор: все совершают случайные ошибки, порой, чтобы найти их требуется немало дополнительного времени.

Разработчик чаще всего оценивает время исключительно на разработку нового функционала, и забывает добавить к этому время на соединение нового модуля с основным проектом или же написание тестов на новый код, документирование и пр. – это приводит к тому, что спустя какое-то время придется вернуться к этой задаче; усложняет процесс планирования и портит статистику, а также отдаляет реальное положение дел от состояния в системе.

Так же выполненная работа может не удовлетворить заказчика. Виноватыми в этой ситуации могут быть как заказчик, так и менеджер проекта или же сам разработчик. Часть информации, при передачи от первого лица к третьему, может исказиться, или быть неправильно интерпретирована. Так случается довольно часто в мелких задачах, не является критическим и есть обычной практикой методологии Agile. Тем не менее, в любом случаи незавершенную задачу следует довести до конца, что подразумевает дополнительную трату времени.

И, несмотря на все эти факторы, после определенного времени обучения, есть вероятность, что система будет давать приблизительные результаты.

Очевидно, что ни одна система не заменит в процессе принятия решения хорошего руководителя, так как невозможно спроецировать в алгоритм все факторы отбора. Человеческий фактор всегда был и останется решающим. Но разрабатываемая система не претендует заменить проект-менеджера. Но когда сотрудников достаточно много, и физически невозможно каждый день контролировать каждого, статистика может привлечь внимание и вовремя отреагировать.

Рассмотрим пример. При добавлении новой задачи, проектный менеджер указывает верхний предел для задачи в определенное количество часов, за которое, по его мнению, задачу наверняка можно выполнить. В случаи, если число часов, в таблице по этой задачи у сотрудника превысит введенную менеджером норму, сработает нотификация. Учитывая тот принцип, что задачи на трекер задач должны добавляться на продолжительность не более восьми часов, это значит, что менеджер будет уже на следующий день уведомлен о выходе за рамки по задачи и сможет принять меры.

Вполне возможно, что в более чем половине случаев подобные подсказки будут малоэффективны. Но основная идея состоит в том, что даже если определенные функции окажутся невостребованными, это никак не нанесет вред процессу, ввиду нулевых затрат на получения этой информации.

### Выводы

Понятие управление проектами связано с необходимостью постоянно наблюдать за их состоянием и выполнением. Управление проектами является неотъемлемой частью повседневной деятельности руководителей разного уровня. Когда над одним проектом трудятся большое количество человек, удобная система управления – единственный способ постоянно контролировать разработку.

Использование подходящих средств позволяет более полно учитывать проектные риски, оптимизировать использование имеющихся ресурсов и избегать конфликтных ситуаций, где могла бы возникнуть путаница между разработчиками, и контролировать исполнение составленного плана.

### Литература

- [1] Эффективность использования информационной системы управления проектами (ИСУП) // <http://project-manager.org.ua/content/view/15/1/>
- [2] Коротко о гибкой методологии разработки Agile // [http://ru.wiki-pedia.org/wiki/Гибкая\\_методология\\_разработки](http://ru.wiki-pedia.org/wiki/Гибкая_методология_разработки)
- [3] XP методология // <http://citforum.ru/SE/project/scrum/>
- [4] Каждому проекту своя методология проектирования и разработки // [http://cit-forum.univ.kiev.ua/SE/project/meth\\_per\\_project/](http://cit-forum.univ.kiev.ua/SE/project/meth_per_project/)
- [5] Записки РМ'а о RedMine // <http://pmant.livejournal.com/18169.html>
- [6] Установка RedMine // <http://logan.pp.ru/node/150>
- [7] Pivotal Tracker – сказка для управления софтверными проектами // <http://www.livebusiness.ru/news/8817/>