

УДК 331.101.3

Ерофеев Егор Валерьевич

ЗАО «Лаборатория Касперского»

Россия, Москва¹

Тестировщик программного обеспечения

E-Mail: Erofeev.sdk@gmail.com

Инновационная мотивация в командобразующих группах для быстрой разработки программного обеспечения

Аннотация: Факт, что мотивация является наиболее важным фактором для повышения производительности команды и качества продукта не является новым открытием.

Несмотря на то, что некоторые исследования мотивации среди работников организации информационных технологий уже проводились, в них явно прослеживается преобладание подхода, направленного на улучшение процессов разработки, нежели внутреннего состояния людей. Современные теории основываются на удовлетворении внутренних потребностей сотрудников, особенно в условиях командной и быстрой разработки.

В этой статье были рассмотрены основные критерии удовлетворенности работой разработчиков программного обеспечения, и доказано, что методы разработки при прямом итеративном взаимодействии с клиентом полностью соответствуют потребностям работников в организации информационных технологий.

Поскольку в настоящее время, большая часть программного обеспечения разрабатывается командами, был взят командный подход, показывающий сильные связи между мотивацией и эффективной коллективной работой, а затем показывает, как методы разработки при постоянном взаимодействии с клиентом положительно связан не только с удовлетворенностью, но и с производительностью.

Был рассмотрен комбинированный способ разработки: быстрой и при итеративной сдаче продукта клиенту и показано, что такой путь является оптимальным для быстрой и качественной разработки программного обеспечения.

Ключевые слова: Мотивация; программист; командный подход; гибкое программирование; внутренняя мотивация; внешние стимулы.

Идентификационный номер статьи в журнале 96EVN114

¹ 123060, Москва, 1-й Волоколамский проезд, 10, строение 1

Egor Erofeev
ZAO «Kaspersky LAB»
Russia, Moscow
E-Mail: Erofeev.sdk@gmail.com

Innovational motivation in teamwork for agile and qualitative development of the software

Annotation: Now no one doubts the fact, that the motivation is a crucial factor to increase the productivity of the team and quality of the product.

Despite of many researches have been made to discover the inner reasons of information technology organization's employee motivation, focus has been set on process-centric methodologies, rather than on intrinsic state of the people. Modern theories are based on satisfaction of the inner necessities of the employees, especially in team and agile development.

In this article the main criterions of the developer's motivation have been listed, and proven then agile programming method is completely correlated with the needs of the employees in the information technology company.

Since nowadays, the most projects are elaborated by teams, it was taken team method expressing strong relation between motivation and collective work, and then, shows how the agile programming is correlated with the motivation and effective collective work, and then shows how agile methods of the development is positively linked with the both motivation and performance.

The combined method of the research was studied: agile with the iterative presentation of the product to client. It was shown that such a way of programming is optimal to quick and qualitative development.

Keywords: Motivation; developer; team methods; agile programming; intrinsic motivation; external stimulus.

Identification number of article 96EVN114

1. Теории мотивации

Несколько теорий было разработано, чтобы попытаться охарактеризовать понятие мотивация. Каждая из них имеет свои сильные и слабые стороны. Однако, ни одна из них не достаточна, чтобы быть всеобъемлющей и способной с успехом применяться в любой ситуации.

Факторы, влияющие на мотивацию, могут быть классифицированы в две основные категории: внутренние факторы и внешние стимулы.

Внутренние факторы исходят от самой работы и целей, устремлений самого работника, то есть, достижения, возможности для роста, социальный статус и т.д.

Внешние факторы являются теми, которые зависят от среды, или основных человеческих потребностей, то есть, зарплата, офисные помещения, ответственность и т.д.

Некоторые известные мотивационные теории, которые могут помочь в объяснении того, что мотивирует программистов являются иерархия потребностей человека по Абрахаму Маслоу, теория Фредерика Герцберга на мотиваторы и демотивирующие факторы [4], и теория мотивационных достижений Дэвида МакКлелланда [9].

Иерархия потребностей человека по Маслоу классифицирует потребности человека на пяти уровнях. В соответствии с этой теорией, потребности более высокого уровня становятся мотиваторами только тогда, когда потребности на нижнем уровне удовлетворены. Иерархия от самого низкого до самого высокого уровня, представляется в таком виде:

- Физиологические, т.е. зарплата, офисные помещения, удобство рабочего места
- Безопасность, то есть, гарантии занятости, пенсионное обеспечение, медицинское страхование, отпуска по болезни
- Социальные, т.е. взаимодействие с коллегами и клиентами, работа в команде
- Самооценка, т.е. репутация, признание и оценка со стороны коллег, подчиненных и руководителей
- Самореализация, т.е. реализация полного потенциала личности, " то, кем человек может быть, кем он должен быть " [7].

Теория мотивации по Герцбергу, рассматривающая мотивирующие и демотивирующие факторы, опирается на различные предположения из теории Маслоу. В соответствии с ней, есть факторы, которые оказывают положительное влияние на повышение мотивации – на мотиваторы, которые Герцберг отождествляет с внутренними факторами; а также другие факторы, которые должны присутствовать во избежание демотивации, но они сами по себе не могут повысить мотивацию – демотивирующие (гигиенические) факторы, которые Герцберг отождествляет с внешними факторами [4].

Согласно этой теории, мотиваторы вытекают из "этой уникальной человеческой характеристики, способности достижения, а через достижение испытывают психологический рост" [4]. Они, в порядке их важности: достижение, признание, собственно работа, ответственность, продвижение, и возможность роста.

Вместо этого, демотивирующие факторы являются следствием животной природы человека и относятся к основным биологическим потребностям, например, потребность в пище делает деньги необходимостью. Некоторыми гигиеническими факторами являются политика компании, офисные помещения, авторский надзор, личная жизнь, и зарплата.

Теория мотивации достижения МакКлелланд характеризует мотивацию конкретного класса людей - те, кто имеет сильное желание достичь. Согласно этой теории, люди, внутренне мотивированные на достижения, имеют следующие характеристики:

- Они предпочитают трудные, но потенциально достижимые цели
- Они предпочитают идти на оправданный риск
- Они больше озабочены личным достижением, чем вознаграждениями за успех
- Они имеют сильную потребность в конкретной работе релевантной обратной связи. Они хотят знать, насколько хорошо они ее выполняют.

Эти две теории связаны друг с другом. Внешние факторы Герцберга соответствуют факторам теории Маслоу, внутренние факторы соответствуют более высокие. Люди замотивированные на достижения имеют тенденцию быть более мотивированными на внутренние факторы Герцберга. Сами достижения является внутренним фактором.

В общем, на рабочем месте, внутренние факторы, как правило, гораздо более эффективны, чем внешние, мотивирующие людей [13].

2. Мотивация процесса разработки программного обеспечения

Первые десять мотивационных факторов для разработчиков программного обеспечения в порядке убывания важности Boehm [2]

- Достижения
- Возможность для роста
- Работа
- Признание
- Прогресс
- Технический надзор
- Ответственность
- Отношения с коллегами
- Отношения с подчиненными
- Зарплата

Данным, на которых основывается список уже более чем 25 лет, но он остается в силе, так как неплохо соответствует современному опыту IT-компаний.

Достижение является самым сильным стимулом для разработчиков программного обеспечения, кроме того, большинство остальных являются также внутренними факторами. Так МакКлелланд и теории Герцберга подходят для объяснения, что мотивирует сотрудников.

Интересно отметить, что фактор, который используют менеджеры, для того чтобы наиболее сильно мотивировать сотрудников - зарплата - занял последнюю позицию. Конечно, деньги важны, но это становится сильным фактором мотивации или демотиватором (в случае недостатка) только тогда, когда рыночная стоимость конкретного программиста является очень высокой или очень низкой.

3. Работа в команде

В настоящее время большинство проектов настолько сложны, что они могут быть выполнены только в команде, так что имеет смысл определить факторы мотивации команды, которые делают ее эффективной.

Ларсон и Ла Фасто [6] провели 3-летнее исследование, чтобы выявить характеристики успешной команды. Команды изучали в диапазоне от футбольных команд вплоть до той, которая спроектировала самолет Boeing 747. Среди них не было команд по разработке программного обеспечения. Они обнаружили, что все высокоэффективные команды всегда обладали следующими характеристиками

- Четко-сформулированная цель
- Ориентированная на результат структура
- Компетентные члены команды
- Единая приверженность
- Благоприятный рабочий климат
- Стандарты качества
- Внешняя поддержка и признание
- Лидерство

Из этого списка видно, что эффективная работа в команде имеет прочную связь с мотивацией.

3.1. Четкая, выполнимая цель

Четкая выполнимая цель является абсолютно необходимым условием для успешного выполнения проекта. Четкость означает, что она возможна, конкретна и однозначна. Примером ясной цели может быть "исполняемый код не должен использовать более 500 МБ оперативной памяти в любой момент времени". Пример плохо сформулированной цели "исполняемый код не должны использовать слишком много оперативной памяти", что, конечно, весьма расплывчато. Цель выполнима, если, с точки зрения команды, она важна и стоит того. Например, это может быть технической проблемой, для выполнения которой требуется приложение всех сил в команде. Люди хотят быть вовлечены в такой проект, который дает им возможность узнать или создать что-то новое, так что если цель ясна, но не выполнимая, достичь ее может быть очень трудно, поскольку она может быть воспринимается как неинтересная или вовсе даже как бесполезная.

3.2. Ориентированная на результат структура

Ориентированная на результат структура выражается в команде, имеющей систему мотивов и стимулов, завязанных на результат, когда она организована в соответствии с целью, что коллектив должен достичь. Структура команды понимает процесс, каналы связи, роли и навыки члены команды. Это является гигиеническим фактором. На самом деле, его присутствие делает достижение возможным, но не мотивирует людей внутренне, и его отсутствие, безусловно, демотивирует, или может сделать достижение намного более трудоёмким, а в худшем случае, невозможным.

3.3. Компетенция

Компетенция имеет важное влияние на мотивацию достижений. Мотивированные на результат люди предпочитают сложные, но потенциально достижимые цели. Недостаточная компетентность может сделать достижение цели невозможным. Есть два типа компетенций, оба одинаково важны: технические компетенции и личностные навыки. Технические компетенции относятся к знаниям и навыкам, необходимым для достижения цели команды. Они абсолютно необходимы. Личные компетенции относятся к личным навыкам личности, а также способность эффективно работать в команде - они могут существенно оказать влияние на производительность команды. Команда отличных разработчиков, которые не могут хорошо работать друг с другом, является как правило, отстающей от команды средних разработчиков, которые хорошо сработались вместе.

3.4. Единая приверженность

Единую приверженность определить не так легко. Это "командный дух", когда люди чувствуют, сильную идентификацию себя самого с командой. Именно тогда, когда все члены команды готовы посвятить время и энергию для достижения общей цели воедино в одном направлении. Именно тогда, когда команда имеет свою собственную идентичность. Единая приверженность можно стимулировать, в первую очередь, путем создания четкой, выполнимой цели. Тогда с участием команды во всех этапах проекта, вовлечение усиливает обязательство. Если унифицированной приверженности не хватает, даже если существует четкая, подъемная цель, то вероятность успеха сильно сокращается.

3.5. Рабочий климат

Рабочий климат описывается фразой "хорошо работать вместе". Важно способствовать единой целеустремленности, чувству принадлежности, и дать членам команды возможность для роста. Для того чтобы иметь совместный климат, необходимо для членов команды, чтобы доверять друг другу. Таким образом, они могут сосредоточиться на достижении цели. Кроме того, коммуникации и координации являются более эффективным, и качество результата значительно улучшается.

3.6. Стандарт качества

Стандарт качества определяет ожидаемый уровень производительности с сохранением нужного уровня качества. Он определяет ожидания от уровней квалификации членов команды, по инициативе и усилиям, которые они могут продемонстрировать, о том, как результаты должны быть достигнуты и т.д. Достижения определяет стандарт, в котором ожидается уровень производительность очень высока. Следствием установления высоких стандартов является то, что ожидания от команды очень высоки. Это дополнительный стимул, который может заставить членов оказать давление друг над другом для того, чтобы уложиться в срок и соответствовать качеству. Результатом является прирост самооценки членов команды. Чтобы следовать стандартам необходимо проявлять трудолюбие и большую дисциплину. Они не должны быть указаны в качестве общих принципов, таких как "Код должен быть отличного качества", но они должны быть определены в плане, что может быть сделано конкретно, чтобы следовать им, например, они могут поручить использование модульных тестов, рефакторинга и парное программирование как методы, чтобы сохранить или повысить качество кода.

3.7. Внешняя поддержка

Внешняя поддержка проявляется в предоставлении команде внешних необходимых ресурсы, чтобы работа выполнялась более эффективно. В мотивационной теории, это «гигиенический» фактор. Без достаточной внешней поддержки очень трудно достичь любую цель.

Признанием являются награды, связанные с достижением. Награды и вознаграждения должны быть привязаны к производительности и рассматриваться как соответствующие меры членами команды. Признание является сильным мотиватором для разработчиков программного обеспечения.

3.8. Лидерство

Лидерство является одним из важнейших факторов, определяющих эффективной коллективной работы. Очень эффективный стиль руководства - это тот, что Ларсон и Lafasto называет принципиальным руководством [6], и Хайсмит [5] руководство - сотрудничество. Принципиальные лидеры не отдают приказы, они вдохновляют и влияют на людей, чтобы добиться цели они доверяют своим последователям, и используют власть очень «экономно». Эффективные лидеры, в соответствии с Ларсон и Lafasto [6] " (1) создают видение; (2) создают изменение, и (3) раскрывают талант ". В команде с такого рода лидера есть отличная возможность для ответственности, техническая.

В заключение все эти факторы эффективной сыгранности сильно связаны друг с другом. Большинство характеристик мотиваторов или гигиенических факторов влияют на эффективность команды, а остальные имеют прямое влияние на эти же характеристики и факторы.

4. Гибкая разработка и командная работа

В этом разделе показано, какие основные характеристики имеют гибкие методы разработки, которые делают возможной эффективное сотрудничество в команде.

4.1. Четкие выполнимые цели

Пошаговая, итеративная разработка наряду с успешным сотрудничеством программистов с пользователями играет центральную роль в поддержании четкости и выполнимости цели.

Использование итеративной разработки позволяет программистам и клиентам определить и установить небольшие, но четкие цели. Важной частью итеративности является прямо-сдаточные тестирования. Их заключения действительно делают цель ясной для всей команды.

Использование, предпочтительно небольших итераций позволит команде получить обратную связь, необходимую для правильной оценки проделанной работы, соответствует ли она тому, что ожидает клиент. На самом деле, прямо-сдаточные тестирование очень полезны, к тому же опыт использования программного обеспечения дает клиенту лучшее понимание его собственных требований. Это, зачастую, приводит к уточнению цели без потери ясности.

Постановка выполнимых целей является сложной задачей. Не существует готовых рецептов для ее решения. Конечно, если клиент тесно работает с командой, он может оказать

большую помощь. На самом деле, он может непрерывно напоминать о (большой) важности программного обеспечения для нее или ее компании.

Некоторые часто успешные методы, чтобы создать задачу путем задания жестких, но достижимых сроков или дать команде возможность получения новых навыков.

4.2. Структура разработки, ориентированная на результат

Гибкие команды структурированы, чтобы завершать комплексное программное обеспечение в срок и в рамках бюджета в контексте частых изменения в требованиях.

Эффективная структура команды имеет четыре необходимые функции [6].

Во-первых, есть четкие роли и ответственность. Для гибкой разработки некоторых из них означает прав, которые закреплены за клиентом и командой разработчиков.

Клиент имеет следующие права:

- Право получить общую архитектуру, которое определяет, что может быть достигнуто, когда и за какую стоимость
- Право на получение максимального качества для каждой итерации
- Право менять приоритеты, не неся в непомерные расходы
- Право быть своевременно проинформированными об изменениях в плане
- Разработчики имеют следующие права
- Право на четкие требования с четкими приоритетами
- Право всегда разрабатывать качественный код
- Право запрашивать и получать помощь от своих коллег, своих менеджеров и клиентов
- Право создавать свои собственные оценки и обновлять их, когда проблема становится все более точно определённой
- Право выбирать свои обязанности, а не иметь их, насильно возложенные на них

Роли и ответственность внутри команды разработчиков зависят от конкретной используемой методологии.

Во-вторых, существует эффективная система обратной связи. Гибкая разработка делает акцент на вербальную коммуникацию между сотрудниками - возможно, самый эффективный канал связи между людьми; члены команды, как правило, расположены близко друг от друга, возможно, в одной и той же комнате – так, скорость передачи данных оптимизирована. Клиент «поощряется» тесно взаимодействовать с разработчиками - так петля обратной связи сокращается, и цель проекта остается видимой и ясной.

В-третьих, есть способ для мониторинга индивидуальной работы и обеспечение обратной связи. В гибком программировании это следствие высокого уровня взаимодействия между заинтересованными сторонами. Если кто-то не делает все возможное, зависящее от него, это становится ясно сразу для всех в команде.

В-четвертых, все суждения должны быть на основе фактов. В методе гибкой разработки все виды деятельности и коммуникаций, в том числе процессы разработки

программного обеспечения – проходят тест «полезности»: они должны способствовать каким-то образом достижению цели, в противном случае они отбрасываются. Процесс упрощения - выполнять только мероприятия, которые упрощают работу команды. Документация пишется, только если есть люди, которые собираются ее читать. Программное обеспечение хранится в наиболее простом виде, так чтобы было легче изменить. Будущие расширения функционала будут рассмотрены, когда возникнет такая необходимость.

Все эти методы позволяют команде разработчиков достигать целей и сосредоточиться только на том, что имеет значение для технического решения задачи.

4.3. Компетентные Члены Команды

Методу гибкой разработки нужны люди как с техническими, так и с личными навыками. Последние являются как раз наиболее важными. На самом деле, способность овладевать новыми навыками, чтобы адаптироваться к меняющейся ситуации, и применить приобретенные навыки по-новому, для гибкой разработки программного обеспечения гораздо важнее, чем наличие сильных технических навыков. Кроме того, высокий уровень взаимодействия требует гибких методов, что требует от людей способности эффективно сотрудничать с другими.

4.4. Единые обязательства

Гибкие методы как правило, связывают всю команду на всех этапах развития. Клиент находится в тесном контакте с командой программистов, так что каждый может лучше понять требования и цель. Собrania по поводу архитектуры продукта широко используют методы, такие как CRC-карты и доски - что привести всю команду вместе, чтобы обсудить архитектуру и вопросы программирования, в котором каждый может внести вклад. Все эти вещи помогают значительно в содействии становления единой приверженности.

4.5. Рабочий Климат

Гибкие методы уделяют особое внимание сотрудничеству. Первый пункт в Agile Manifesto четко утверждает, что люди и их взаимодействия более ценны, чем процессы и инструменты, их сотрудничество с клиентом предпочтительнее контрактной работы. Установка совместного климата - одна из главных целей некоторых гибких практик программирования. Акцент ставится на непосредственную коммуникацию, одно или, по крайней мере, очень близкорасположенные офисные помещения, коллективная собственность кода и сотрудничество с клиентами, пошаговая разработка,- все эти факторы, которые помогают создать атмосферу доверия и сотрудничества.

4.6. Стандарты качества

Уровень реализации, к которому стремятся методы быстрого программирования достаточно высок: удовлетворенность клиентов в условиях разработки непрерывности во времени, своевременности и на ограниченно-бюджетной основе предоставление ценного программного обеспечения.

Чтобы облегчить соответствие таким стандартам, гибкие методы используют некоторые конкретные практик, такие как постепенно, пошаговое развитие, всеобъемлющее тестирования, стандарты кодирования.

Некоторые методы - например, Extreme Programming (XP) - также используют парное программирование и содействуют концепции коллективной собственности кода. Эти две

технологии являются мощным стимулом идти в ногу с установленными стандартами, так как каждая строка кода может быть, потенциально понятна любому члену команды.

4.7. Внешняя поддержка и признание

Гибкие методы четко осознают важность внешней поддержки. На самом деле один из принципов Agile Manifesto утверждает, что "разрабатывать проекты в среде мотивированных людей. Дайте им среду и поддержку, в которой они нуждаются, и доверяйте им, чтобы получить код, соответствующий всем стандартам. " Признается, что без соответствующих ресурсов, разработка программного обеспечения просто невозможна.

Распределение офисных комнат, расположив разработчиков в близко друг с другом, акцент на вербальном общении, наличие соответствующих средств разработки, и тесное сотрудничество с клиентом требуют немало внешней поддержки.

Когда признание исходит от довольного клиента, оно имеет мощный эффект. Простой акт публичного демонстрирования признания, будь то похвала или вручение премии за хорошо выполненную работу является очень мощным мотиватором. Это помогает в повышении самооценки разработчиков и уровне доверия между ними и заказчиком. Это, в свою очередь, приводит к лучшей коммуникации и лучшего программного обеспечения.

Использование итеративной разработки может сыграть важную роль в увеличении внешней поддержки и признании. Успешная команда, которая обеспечивает результаты на ранних стадиях проекта и непрерывно их поддерживает и развивает в течение всего цикла задачи, скорее всего, получит необходимую ей поддержку от каждой из заинтересованных сторон.

5. Вывод

Было показано, почему гибкие методы могут быть более продуктивными и привлекательными для разработчиков. Они используют самый важный фактор для производительности и морального состояния работников: мотивацию. Роль процессов и инструментов в данном контексте по-прежнему очень важна, так как они используются для упорядочения все повторяющихся задач, позволяя разработчикам сосредоточиться на том, что действительно имеет значение: выпуск качественного программного обеспечения, соответствующего всем требованиям клиентов.

ЛИТЕРАТУРА (REFERENCES)

1. Beck, K., et al., *The Agile Manifesto* , <http://www.agilemanifesto.org>
2. Boehm B. W., *Software Engineering Economics*, Prentice Hall, 1981
3. Cockburn, A., *Agile Software Development*, Addison Wesley, 2002
4. Herzberg, F., *One More Time: How do you Motivate Employees?*, Harvard Business Review, 1968
5. Highsmith, J., *Agile Software Development Ecosystems*, Addison Wesley, 2002
6. Larson, C., E., LaFasto, F., M., *Teamwork: what must go right / what can go wrong*, Sage Publications, 1989
7. Maslow, A., *Motivation and Personality*, Addison Wesley, 1987
8. Mayo, E., *The Human Problems of an Industrial Civilization*, Macmillan, 1933
9. McClelland, D., *The Achieving Society*, The Free Press, 1967
10. Peters, J., Waterman, R., H., *In Search of Excellence: Lessons from America's Best Run Companies*, Harper, 1982
11. Poppendiek, M., Poppendiek, T., *Lean Software Development*, Addison Wesley, 2003
12. Taylor, F., W., *The Principles of Scientific Management*, Dover Publications, 1998
13. Thomas, K., W., *Intrinsic Motivation at Work: Building Energy & Commitment*, Berrett-Koehler, 2003
14. Weinberg, G., M., *The Psychology of Computer Programming: Silver Anniversary Edition*, Dorset House Publishing, 1998
15. http://ru.wikipedia.org/wiki/%C3%E8%E1%EA%E0%FF_%EC%E5%F2%EE%E4%EE%EB%EE%E3%E8%FF_%F0%E0%E7%F0%E0%E1%EE%F2%EA%E8

Рецензент: Вашаломидзе Елена Валерьевна, кандидат экономических наук, НИИ Труда и социального страхования.