

Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники

УДК _____

Масловский
Олег Викторович

Повышение эффективности разработчиков программного обеспечения

АВТОРЕФЕРАТ

на соискание степени магистра технических наук
по специальности 1-23 80 08 Психология труда, инженерная психология,
эргономика

Научный руководитель доктор
технических наук,
профессор А. М. Лазаренков

Минск 2015

Введение

Современный мир требует постоянно подстраиваться под него и изменяться абсолютно всем отраслям человеческой жизни. Постоянные новшества, вносимые в человеческую жизнь, уже никого не удивляют, а лишь заставляют перестраиваться и идти в ногу со временем. Особенно заметно постоянное изменение, совершенствование при разработке различного рода программного обеспечения, которое должно обеспечить постоянно растущие потребности пользователей на различных устройствах, быть эргономичным, быстрым и конкурентоспособным.

Вследствие постоянного появления всё новых и новых технологий в сфере информационных технологий для написания программы, которая будет удовлетворять всем требованиям современного мира программист должен постоянно совершенствовать свои знания, используя новые методики разработки программного обеспечения.

Однако для успешного выполнения ИТ-проекта недостаточно выбрать эффективные технологии и средства разработки, обеспечить необходимый бюджет и найти квалифицированных разработчиков. Помимо этого необходим еще один очень важный критерий, речь о котором пойдет ниже.

Как известно, в любой организации существуют правила и методики, по которым участники проекта (заказчики, бизнес-аналитики, разработчики, тестировщики, технические писатели, менеджеры) распределяют между собой задачи, взаимодействуют друг с другом, создают проектные артефакты (спецификации, исходный код, документацию). Эти правила могут быть четко организованными или хаотичными, быть формально документированными или существовать в головах проектной команды, но в любом случае именно их совокупность называется процессом разработки. И именно он влияет на конечный результат продукта не меньше, чем остальные факторы в своей совокупности. Ибо если будут найдены замечательные высококвалифицированные разработчики, будут иметься средства на разработку проекта и будут использоваться эффективные технологии, но при этом члены команды не будут взаимодействовать друг с другом, процесс разработки не будет налажен, то ожидать блестящий конечный результат от, казалось бы, безупречного начала не стоит.

Общая характеристика работы

Разработка программного обеспечения, несмотря на обилие методологий и исследований, до сих пор, является слабо изученной областью. Нельзя однозначно сказать, как правильно создать универсальную команду для разработки программного обеспечения, кто из программистов и

в каком соотношении должен в нее входить, как организовать взаимосвязь и взаимодействие между разработчиками, как осуществлять сам процесс разработки, чтобы в конечном итоге мы могли получить качественный продукт при минимальной затрате времени и усилий. Нет нигде ярко выраженной информации о том, как создать наиболее эффективную команду разработчиков. Поэтому в данной работе мы постараемся выяснить такую малоизученную область как эффективность взаимодействия разработчиков программного обеспечения при разработке различного программного обеспечения.

Исходя из этого целью данной работы является повышение эффективности взаимодействия разработчиков программного обеспечения.

Объект исследования работы – непосредственно разработчик программного обеспечения.

Предмет исследования – взаимодействие разработчиков программного обеспечения как внутри команды, так и за ее пределами.

В ходе исследования предполагается рассмотреть схемы взаимодействия разработчиков при выполнении проектов на примере команд компании Itransition и по итогу выбрать наиболее эффективную.

Данная работа состоит из введения, трех глав и заключения. Первая глава посвящена общим теоретическим вопросам разработки программного обеспечения, а также проблемам, с которыми сталкиваются программисты при его разработке. Во второй главе речь пойдет непосредственно о взаимодействии программистов различных типов как внутри команды, так и за ее пределами. В качестве примера взято двухлетнее наблюдение за командами компании Itransition. Третья глава включает в себя рекомендации по созданию команд с максимально эффективным взаимодействием разработчиков, а также пути решения вопроса о повышении эффективности работы разработчиков в таких командах. И, наконец, в заключении будут изложены основные выводы по проблемам в выбранной для исследования теме.

Краткое содержание работы.

В первой части работы был обзор литературы о текущей ситуации в мире разработки программного обеспечения, были озвучены существующие проблемы эффективности разработки программного обеспечения а также были поставлены задачи на исследование. Из главы становится ясно что, в настоящее время имеется пять основных процессов (методов) разработки программного обеспечения, в зависимости

от которого меняется взаимодействие разработчиков программного обеспечения. Каждая из методологий имеет свой круг применения и свои достоинства и недостатки.

Однако только выбором правильной методологии нельзя решить проблему эффективности разработки программного обеспечения и устранить все проблемы, возникающие при этом. В литературе не содержится прямых рекомендаций для повышения эффективности работы команды при разработке программного обеспечения. В данной главе мы выявили и проанализировали основные ошибки, допускаемые командами при разработке программ.

Во второй части были описаны исследования которые проводились в компании Itransition, была исследована личность программиста, также было предложено деление программистов в зависимости от способностей (тимлид (TeamLead), разработчик (developer), архитектор).

В главе было рассмотрена необходимость взаимодействия всех участников разработки программного обеспечения для получения качественного и конкурентоспособного программного обеспечения. Способы взаимодействия между разработчиками программного обеспечения зависят от типа (методологии) разработки программного обеспечения. При водопаде идет взаимодействие между тимлидом и разработчиками, тимлидом и архитектором, тимлидом, архитектором и заказчиком. Преобладает живое общение. При итеративной разработке, в отличие от водопада, больше взаимодействия между самими разработчиками как через код или специально созданные системы, так и живое общение внутри команды. При RUP немало взаимодействия идет через документооборот, спецификации что непривычно и зачастую не нравится разработчикам. Такая методология подходит для глобальных проектов. Взаимодействие между разработчиками при использовании гибких методологий идет активное взаимодействие между всей командой. Причем тут может быть, как живое общение, так и переписка, звонки. При этом нельзя забывать и о межкомандном взаимодействии, возникающем при выполнении одного проекта несколькими командами, при обмене команд опытом. Это необходимо для сохранения проекта в едином стиле, распределению задач и контролем за их выполнением между разными командами, обсуждению ошибок и путей их решения (предотвращения) на проекте и т.д. Поэтому для повышения коммунибельности программистов кампании проводят различные мероприятия (тимбилдинги, корпоративы и т.д.).

Также немалое значение на модель и способ взаимодействия оказывает личность разработчика. При формировании команды стоит учитывать и

личности разработчиков, в нее входящих. В зависимости от обладания теми или иными личностными качествами каждый из разработчиков может стать определенным типом разработчика. Так, разработчиком (в узком смысле слова) джуниор может стать при наличии усидчивости, эрудиции, способности и стремлению к постоянному изучению чего-то нового, внимательности, абстрактного и системного мышления. Стать тимлидом можно при наличии сильной, волевой личности, умении повести за собой команду, ответственности, явных лидерских задатков, познаний в психологии, коммуникабельности, умении распределять задачи между разработчиками, стрессоустойчивости. Также тимлид должен быть одновременно и хорошим техническим специалистом, и менеджером. А стать архитектором – при наличии немалой усидчивости, логического мышления, стремлении постоянно познавать что-то новое, скрупулёзности, дотошности, умении хранить и обрабатывать большие объемы информации.

В третьей части работы были предоставлены пути повышения эффективности разработчиков программного обеспечения.

Наиболее оптимальная команда для разработки программного обеспечения состоит из шести человек, включая тестировщика, и находится под управлением тимлида. На несколько таких команд (или же на одно небольшое подразделение) должен приходиться архитектор. Команда без тимлида становится менее управляемой и ответственной, что сказывается на сроках и качестве выпускаемого программного обеспечения.

Также на создание оптимальной команды оказывает влияние цель ее создания и проекты, на которых она будет работать. Так, для динамичных, недолгих проектах, использующих передовые технологии в разработке программного обеспечения, стоит брать умных специалистов, готовых к самообразованию и желающих хорошо поработать. Для длинных проектов, использующих годами одни и те же технологии обычно формируются команды из более ленивых и менее перспективных программистов, не заботящихся о дальнейшем карьерном росте. На такие проекты не стоит опирать ценных сотрудников, иначе они частично утратят свою профессиональную пригодность.

Не стоит забывать и о новеньких программистах в команде. Чтобы из джуниора «вырос» полноценный квалифицированный специалист, в командах практикуется плавный способ внедрения нового разработчика, при котором задачи для джуниора усложняются с увеличением его опыта, а более опытные коллеги на первых порах помогают новичку советами, а также рекомендуют материал для саморазвития. Но самосовершенствование в профессиональной среде касается не только молодых программистов.

Изучать новые технологии в программировании должны все без исключения разработчики. И ключевую роль здесь играет самообразование. Однако поскольку оно довольно трудоемко, то помимо этого используются и другие, воспринимаемые разработчиками больше в качестве отдыха, чем в качестве познавательного материала: различные открытые конференции разработчиков (в том числе и международные), форумы, чтение и прослушивание лекций, курсы повышения квалификации. Для молодых программистов также в целях повышения их самооценки и значимости в команде устраиваются поощрительные конкурсы (лучший разработчик месяца, сезона и т.д.).

Все это помогает разработчикам программного обеспечения оставаться высококвалифицированными специалистами и, следовательно, выпускать качественное программное обеспечение.

Заключение

В работе были предоставлены конкретные советы по повышению эффективности разработчиков программного обеспечения.

Чтобы по итогу разработки получилось программное обеспечение, отвечающее всем требованиям заказчика, являющееся конкурентоспособным, технически правильным, стабильно работающим, удобным в использовании, имеющим красивый вид, богатый функционал и быстрое действие, к процессу разработки стоит подходить с должным вниманием.

Также важно сформировать правильную команду для разработки той или иной программы. Причем на формирование команды оказывает прямое влияние проект, над которым она будет работать, его сложность, срок, на который планируется создать команду (только на этот проект или же на несколько проектов), срок действия проекта (проектов), вид управления командой и т.п. Исходя из этих показателей берутся пропорции программистов различных видов и опыта работы. Если же взять, к примеру, команду, которая будет заниматься постоянно разработкой небольших (до года) проектов, то наиболее оптимальным вариантом такой команды является команда из шести разработчиков, включая тестировщиков, находящаяся под управлением тимлида. На несколько таких команд или небольшое подразделение полагается наличие архитектора.

Не стоит забывать и про взаимодействие разработчиков программного обеспечения между собой, причем как на уровне команды, так и за ее пределами. Стоит отметить, что при каждом из методов разработки программного обеспечения будет иметься свой вариант налаживания взаимодействия в команде и способ общения (живое, переписка, звонки и

т.д.). Но главным является тот факт, что команда должна действовать как единый органичный живой организм, постоянно обмениваясь сведениями и изменениями, тогда и итоги работы такой команды будут плодотворны.

В целях повышения коммуникативных навыков такой необщительной категории людей как программисты, а также для улучшения их взаимодействия между собой и формирования здоровой атмосферы в коллективе в целом кампании организуют различные мероприятия: корпоративы, тимбилдинги, лекции и дискуссии, открытые конференции разработчиков, организует совместный отдых разработчиков и многое другое.

Также стоит уделять большое внимание повышению квалификации разработчиков. Наиболее оптимальным способом вливания нового разработчика в команду является плановый способ, при котором новичку дают вначале более легкие задачи, постепенно увеличивая уровень их сложности, а более опытные коллеги контролируют, делятся опытом и на первых порах помогают советом или подсказкой, а также рекомендуют источники информации для самообразования. В результате такой подготовки из джуниора быстро вырастает мощный разработчик, способный к грамотному обучению новичков.

Соблюдая все эти критерии можно значительно повысить эффективность разработки программного обеспечения и получить действительно качественный и конкурентоспособный продукт, при этом рационально расходуя время и ресурсы разработчиков и используя необходимые технологии.