МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федерально автономное образовательное учреждение высшего образования
«Севастопольский государственный университет»
кафедра Информационных систем

Куркчи Ариф Эрнестович

Институт информационных технологий и управления в технических системах
курс 4 группа ИС/б-41-о
09.03.02 Информационные системы и технологии (уровень бакалавриата)

Реферат
по дисциплине «Управление IT проектами»
на тему «Управление изменениями. Стратегии управления изменениями»

Отметка о зачете _____ _____
(дата)

Руководитель практикума

старший преподаватель	_____	Смирнова Н.Б.
(должность)	(подпись)	(инициалы, фамилия)

Севастополь 2017

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Изменения, запрашиваемые в ходе выполнения проекта, являются неизбежной частью любого проекта. Есть несколько причин неизбежности изменений требований. Среди них внутренние факторы, которые мы можем контролировать, и внешние, которые не подвластны контролю со стороны разработчиков и пользователей.

Внешними факторами являются те источники изменений, которые команда проекта не может контролировать. Вне зависимости от того, как команда разработчиков справляется с ними, необходимо подготовиться технически, эмоционально и методологически, чтобы воспринять эти изменения как часть нормального развития событий при разработке программного обеспечения.

ПРОЦЕСС УПРАВЛЕНИЯ ИЗМЕНЕНИЯМИ

Поскольку изменения являются неотъемлемой частью процесса, а источники их поступления могут быть как внешними, так и внутренними, необходим процесс управления изменениями требований. Такой процесс дисциплинирует команду, позволяет ей выявлять изменения, производить анализ их воздействия и систематическим способом включать те из них, которые наиболее необходимы и приемлемы для системы. Процесс обработки изменений должен включать в себя следующие шаги:

1. Осознать, что изменения неизбежны, и разработать план управления изменениями.
2. Сформировать базовый уровень требований.
3. Установить единый канал контроля изменений.
4. Использовать систему контроля изменений для их фиксации.
5. Обрабатывать изменения по иерархическому принципу.

Рассмотрим более подробно каждый из этих шагов.

1. Осознать, что изменения неизбежны, и разработать план управления изменениями.

Команда должна признать, что изменения требований к системе неизбежны и даже необходимы. Изменения будут возникать, и команда, зная об этом, должна разработать соответствующий план управления изменениями, который разрешит внесение определенных изменений в исходный базовый уровень. Легитимными являются все изменения, так как исходят от заинтересованных лиц, которые имеют как реальную потребность, так и возможность внести реальный вклад в приложение.

2. Формирование базового уровня требований.

Ближе к концу фазы исследования жизненного цикла разработки команда должна скомпоновать все известные требования к системе. Процесс формирования базового уровня может заключаться просто в наложении контроля исправлений на соответствующие артефакты (документ-концепцию, программные требования, модели прецедентов) и публикации базового уровня для команды разработчиков. Собранные в этих документах отдельные требования создают базовый уровень информации о требованиях и предполагаемых прецедентах системы. Этот простой шаг позволяет команде различать известные (“старые”) требования и новые (те, которые были добавлены, удалены или модифицированы). После задания базового уровня гораздо легче выявлять и обрабатывать новые требования. Запрос на новое требование можно сравнить с существующей базой и определить, где оно будет размещаться и не будет ли оно конфликтовать с другими требованиями.

3. Установление единого канал контроля изменений.

Появление новой функции может оказать существенное влияние на требования к программному обеспечению, системную архитектуру, планы тестов и т.д., простое изменение кода может вызвать непредвиденные последствия, иногда даже катастрофические. Кроме того, предлагаемая новая функция может устранять важную будущую функцию системы или затруднять ее реализацию (т.е. воздействие распространяется не только на текущую версию). Есть также трудности, связанные с графиком и бюджетом проекта, за которые отвечает руководство. Пожелания заказчиков о внесении изменений не предполагают официального изменения графика и бюджета, и следует инициировать процесс переговоров до того, как изменение будет принято.

Пожелания заказчиков о внесении изменений не предполагают официального изменения графика и бюджета.

Таким образом, очень важно, чтобы все изменения поступали по одному каналу, чтобы определить их воздействие на систему и принять официальное решение, стоит ли вносить это изменение в систему вообще.

4. Использование систему контроля изменений для их фиксации.

Внешние изменения, которые производятся по запросу клиента, легко выявлять, и они будут естественным образом включены в проект руководством или органом, осуществляющим контроль за изменениями. Но во время разработки возникает огромное множество иных изменений системы. Многие предлагаемые изменения, возникающие во время проектирования, кодирования и тестирования системы, могут казаться не связанными с требованиями (например, исправление ошибок кода или проектирования). Тем не менее необходимо оценить их воздействие. А если подходит срок сдачи, следует даже принять сознательное решение о том, какие ошибки оставить в системе (из-за того, что их исправление может дестабилизировать систему в целом и тем самым поставить под угрозу дату сдачи), а какие — устранить. Помимо этого, многие ошибки могут влиять на требования, вызывать необходимость их согласования или устранения неоднозначности отдельного известного требования. Команде разработчиков придется принять сознательное решение о том, какие недостатки оставить в системе.

В любом случае необходимо проанализировать ситуацию, а также принять решение о том, где изменение будет реализовано в иерархии документов. Следовательно, команде необходимо разработать формальный метод фиксации всех запрашиваемых изменений системы. Это может осуществляться с помощью системы отслеживания изменений и неполадок, которая обеспечивает создание централизованного архива запросов, web-ввод

элементов из любого физического местоположения, автоматическое отслеживание состояния, автоматическую отметку затрагиваемых частей, а также механизм передачи запросов изменений в систему управления требованиями, если это необходимо. Эту систему следует использовать, чтобы фиксировать все предложения и передавать их руководству Совета по контролю за изменениями (ССВ) для принятия решения.

5. Иерархическое управление изменениями.

Проблемой является то, что изменения могут не документироваться и не анализироваться. Если их тщательным образом не обрабатывать, это может привести к неприятным по- следствиям. Изменение одного требования может оказать возмущающее воздействие на связанные с ним требования, проектирование или другие подсистемы. Каждая новая функция/требование оказывает влияние на стоимость, график, надежность и риск, связанные с проектом.

Чтобы ослабить возмущающий эффект изменений требований, необходимо выполнять их в иерархии нисходящим образом. Изменения базового уровня документа-концепции можно отражать в отдельном документе Delta Vision, который, как правило, является совсем небольшим подмножеством исходного документа. Но так как изменение документа-концепции может заключаться в удалении функций, может понадобиться создать совершенно новый исходный набор требований к программному обеспечению, а это может привести к соответствующим изменениям проектирования, кода и планов тестирования.

Хорошим решением является воспользоваться поддержкой автоматических средств, нисходящее распространение возмущения будет отражено механизмом трассировки, который используется при построении пирамиды требований.

Это позволит работать с пирамидой сверху вниз, внося дальнейшие изменения там, где необходимо. Каждое последующее изменение обнаруживает дополнительные “подозрительные связи” или точки более нижнего уровня пирамиды, где необходимо провести дополнительный анализ.

Таким образом, изменение распространяется по иерархии логичным и контролируемым образом.

ЗАКЛЮЧЕНИЕ

Безусловно, по мере развития проекта требования будут меняться, но эти изменения не обязательно дестабилизируют процесс разработки. Если создан всеобъемлющий процесс контроля изменений, а артефакты требований подконтрольны используемой командой разработчиков системе управления конфигурацией, команда будет хорошо подготовлена к решению основных задач управления изменениями.

Важно понимать, что управление изменениями в крупном проекте обычно является слишком объемной задачей, чтобы ее можно было решать вручную. Необходим процесс, чтобы контролировать, каким образом изменения попадают в проект. Также необходимы автоматические средства для того, чтобы понять разветвления изменения, т.е. найти все затронутые им элементы проекта.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Управление изменениями (ITSM) [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Управление_Изменениями_(ITSM)
2. Ашмарина С.И. Управление изменениями: учеб. пособие / 1. С.И. Ашмарина, Б.Н. Герасимов. М.: Рид Групп, 2011. 208 с.
3. Оркина Е.А. Управление изменениями / Е.А. Оркина. Ростов 6. н/Д: Феникс, 2014. 190 с.
4. Шермет М.А. Управление изменениями: учеб. пособие. М.: 10. ИД «Дело» РАНХиГС, 2012. 128 с.