A large pile of colorful LEGO bricks and pieces, including various shapes like plates, beams, and gears, in shades of red, blue, yellow, white, and grey.

# Управление требованиями (ч.2)

# План

1. Оценка задач;
2. Трассируемость требований;
3. Управление изменениями;
4. Домой;
5. Вопросы?

# Оценка задач



# Оценка задач | Определение

Оценка задач – процесс, главной целью которого является определение «затрат» или «стоимости» выполнения задачи. Данный процесс проводится **ДО** непосредственного выполнения задачи, так как это необходимо для прогнозирования сроков поставки задач. Существует множество техник оценки задач. Чаще всего оценивают:

- Временные затраты: часы, дни (линейные методологии);
- Сложность: story points, абстрактные единицы (гибкие методологии);



# Оценка задач | Оценка в story points

## Тезисы:

- Story points – оценка сложности одной задачи относительно другой;
- Такая оценка не имеет выражения во времени;
- Для того, чтобы произвести первичную и последующие оценки, выбирают минимальную единицу производства и в дальнейшем используют ее, как эталонную;
- Должна проводиться по единой шкале (как правило, ряд Фибоначчи).

# Оценка задач | Источники информации для оценок

## Аналогичные задачи

Принцип: задача оценивается по аналогии с другой задачей. Ищется похожая задача и оцениваемой ставится такая же оценка.

## Исторические данные

Принцип: исследуются выполненные похожие задачи и текущие оцениваются по факту прошлых задач.

## Экспертные оценки

Принцип: задача оценивается по результатам обсуждения исполнителей и приглашенных экспертов.

- От большего к меньшему/от меньшего к большему;
- Историческая оценка;
- Параметрическая оценка;
- Мозговой штурм;
- Покер планирования.

# Оценка задач | Пример



# Оценка задач | Пример



- Определим наиболее простую задачу;  
-> *Посадить цветочек. Считаем как 1 sp.*
- Насколько сложнее посадить куст?  
-> *Примерно, как 5 цветочков или 5 sp.*
- Насколько сложнее посадить дерево?  
-> *Примерно, как 3 куста или 13sp.*
- Оценим посадку кактуса.  
-> *Примерно, как куст, однако может оказаться сложнее из-за особенностей растения (колючее). Нужно заложить риск: 8 sp.*

# Оценка задач | Пример



- Поставить горшок с цветами.  
*>> Задача включает несколько другие проблемы, ежели просто посадить цветок. Нужно проанализировать: горшок уже закуплен или нет? Если да, то он точно нам подойдет? Если нет, то есть риск, что нужно будет его адаптировать под наши нужды, соотв. оценка вырастет. А если не куплен, то нам нужно делать его самим, а это уже совсем другая задача.*
- Сделать фонтан.  
*>> Привлекаем эксперта.*

# Оценка задач | Задачи БА при оценке задач

1. Инициирование и подготовка процессов оценивания (грумминг, планинг);
2. Фасилитация процесса;
3. Уточнение, консультирование, корректировка задач;
4. Привлечение сторонних экспертов;

# Оценка задач | Преимущества и риски

## Преимущества

- Прогнозирование поставок;
- Фокусировка команды на своевременном выполнении;
- Определение недостающей информации в требованиях.

## Риски

- При слабой организации процесса, он может оказаться бесполезным;
- Может быть затратным по времени, однако они будут уменьшаться со временем;
- Не соответствие ожиданий стейкхолдеров и оценок команды разработки.

# Оценка задач | Покер планирования

1. Каждому из команды разработки раздаются карты с оценками;
2. РО/БА берет верхнюю историю из беклога и зачитывает ее;
3. Каждый участник выбирает свою оценку, выбирает карту и кладет ее рубашкой вверх;
4. Участники вскрываются;
5. Если есть разница в оценках, то участник, положивший наименьшую оценку и наибольшую аргументируют ее;
6. После обсуждений, все забирают свои карты и повторяют итерации до тех пор, пока оценки не сойдутся.

# Трасируемость требований



# Трассируемость | Определение

Трассируемость – это свойство требований, при котором легко можно отследить связи между требованиями разных уровней, формами требований (артефактами) и прочими объектами.



# Трассируемость | Виды связей требований

Происхождение	Зависимость	Реализация	Валидация
<p>Одно требование порождает или является источником другом другого требования. <a href="#">Примеры:</a></p> <ul style="list-style-type: none"><li>■ Эпика- US;</li><li>■ Бизнес-требование- пользовательское требование;</li><li>■ Диаграмма процесса – описание.</li></ul>	<p>Бывает двух видов:</p> <ul style="list-style-type: none"><li>■ <a href="#">Блокирующая</a> – реализация одного требования невозможна от другого;</li><li>■ <a href="#">Неблокирующая</a> – реализация одного требования было бы легче с реализацией другого.</li></ul>	<p>Связь требования и программного компонента, его реализующего.</p> <p><a href="#">Пример:</a></p> <ul style="list-style-type: none"><li>■ Спецификация – задача в JIRA;</li><li>■ Баг - Требование</li></ul>	<p>Связь требования и тест-кейса либо критерия приемки.</p>

## Трассируемость | Матрица трассировки

Матрица трассировки – это визуализация связей требований с другими объектами в форме таблицы. Для удобства, все объекты (требования, тест-кейсы и пр.) должны быть уникально пронумерованы. Помимо указанных типов связей, матрица может позволить указать тип отношений объектов между собой:

- Один – к – одному;
- Один – ко – многим;
- Многие – ко – многим.

# Трассируемость | Матрицы трассировки. Примеры

Requirements Traceability Matrix (RTM)			
Business User Requirement	System Requirement	Design Requirement	Document
BR1.	UR1.	SR1.	DD1.
		SR2.	DD1.
		SR3.	DD2.
	UR2.	SR4.	DD3.
	UR3.	SR5.	DD3.
		SR1.	DD1.

# Трассируемость | Матрицы трассировки. Примеры

## Use Cases and Referenced Requirements

<u>UC-1</u>	Sign up for course
<u>UC-2</u>	Drop course
<u>UC-3</u>	View prerequisites
<u>UC-4</u>	Search open sections
<u>UC-5</u>	Get class roster
<u>UC-6</u>	Indicate availability
<u>UC-7</u>	Assign professors to courses
<u>UC-8</u>	Determine course offering
<u>UC-9</u>	Begin a semester

			✓	✓	✓	✓	✓	✓
			✓				✓	
							✓	✓
							✓	
								✓

REQ-1 Access to private data must be restricted.  
REQ-1.1 Access to class data is limited to 10 students.  
REQ-1.2 Access to course data is limited to 50 students minimum.  
REQ-2 Notify professor when roster changes.  
REQ-3 10 student maximum.  
REQ-4 50 student minimum.  
REQ-5 Retrieval time for list of courses shown must be available 24/7.  
REQ-6 System must support up to 500 concurrent users.  
REQ-7 Must support multiple browsers.  
REQ-8 Support printing.

1. JIRA и прочие таск-треккинговые системы;
2. Электронные таблицы (Google SH, Excel и др.);
3. Инструменты ведения документации (кросс-ссылки).

# Трассируемость | Почему это важно?

1. Анализ влияния изменений и ошибок(багов);
2. Видимость зависимостей объектов;
3. Контроль разработки, видимость прогресса;
4. Поиск и поддержка;
5. Уменьшение рисков;
6. Идентификация общих объектов.

# Управление изменениями

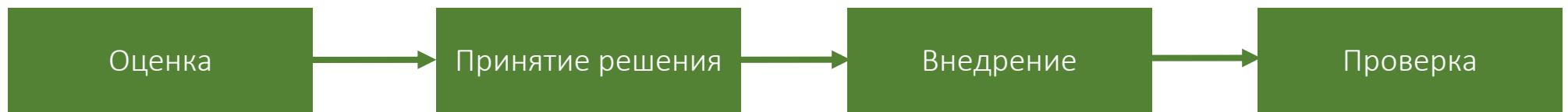


# Управление изменениями | Определение

Управление изменениями – процесс определения, контроля и сопровождения запросов на изменение требований. Основным источником изменений является заказчик либо рынок (продуктовая компания). Является краеугольным процессом в любой компании, так как отсутствие его контроля приведет к большим проблемам, связанных с качеством и своевременностью поставок программного продукта.



# Управление изменениями | Процесс



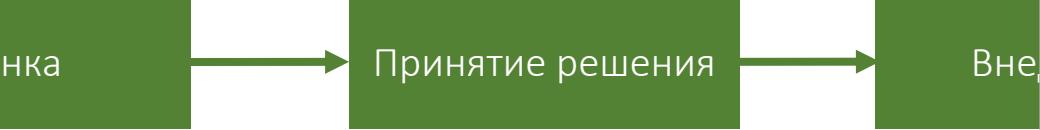
# Управление изменениями | Процесс. Оценка



При поступлении нового запроса на изменение, задача аналитика собрать следующую информацию:

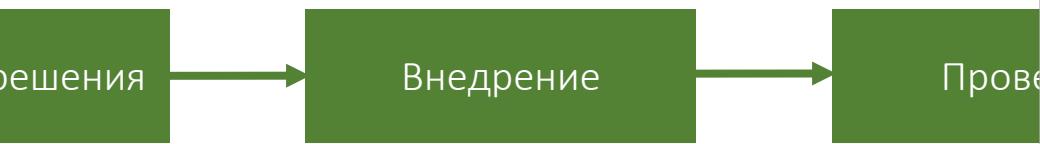
1. Уточнить детали новых требований и их соответствие контракту;
2. Степень воздействия (что заденет);
3. Влияние на проект (вызовет ли задержку поставки);
4. Оценить объем работы (совместно с тех. специалистами);
5. Разработать альтернативные решения и найти способы решения новой задачи существующими средствами.

# Управление изменениями | Процесс. Принятие решения



На данном этапе вы сопоставляете цели проекта, сроки и доступные ресурсы с информацией полученной на предыдущем шаге и предлагаете все возможные варианты внедрения заказчику. Может сложиться такая ситуация, при которой ни одно из предложенных решений не удовлетворяет заказчика, тогда вам необходимо будет эскалировать этот вопрос менеджеру проекта.

# Управление изменениями | Процесс. Внедрение

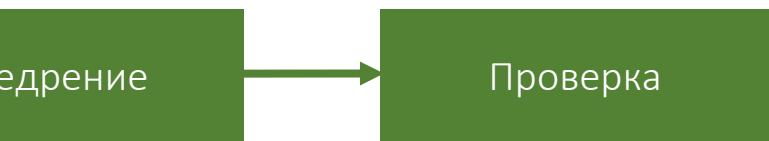


На этапе внедрения задачи аналитика аналогичны его задачам на этапе основной разработки:

- Документирование;
- Актуализация всех связанных артефактов;
- Обновление матрицы трассировки;
- Сопровождение команды разработки;
- Приемка.

В документации может быть краеугольным отметки требований До и После изменения.

# Управление изменениями | Процесс. Проверка



На этапе проверки задача тестировщиков и разработчиков исправить все возникшие ошибки, задача аналитика помочь им в этом, так как, скорее всего, ошибки будут связаны с изменением поведения системы, к которым некоторые ее компоненты были не готовы. После завершения всех этих задач, задача аналитика уведомить заказчика об изменении.

## Тезисы:

1. Для эффективного управления изменениями ваши требования должны быть трассируемыми;
2. Часто следуемая методология разработки имеет описание процесса управления изменениями;
3. В любом случае, процесс должен быть известен и согласован со всеми и документирован;
4. Вы всегда должны быть открыты к изменениям, но в редких случаях может понадобится умение сказать «нет»;
5. Мониторьте количество поступающих изменений, если оно растет, то, возможно, есть глобальная проблема.

# Домой

- Читать: ВАВОК (главы 5, 7, 8)
- Читать: Вигерс «Разработка требований к ПО» (главы 18, 19, 20)

