## Заметки/тезисы к презентации.

### Вводная информация

* Представляется не методология разработки, а способ автоматизации и оптимизации процесса. Ближе всего данный способ к «гибким» методологиям.
* Работает со всеми методологиями (по ГОСТ – водопад, и по современным – гибким (agile)). По ГОСТу – внутри каждого этапа водопада – эджайл, если весь процесс эджайл – ок.

### Слайд 1. Бизнес-цели.

* У любой организации есть бизнес-цели. Описание наших.
* Все процессы в организации должны коррелировать с данными бизнес-целями. Если нет – процессы нужно отменять или улучшать.

### Слайд 2. Основной бизнес-процесс.

* Основной процесс для достижения бизнес целей. Описание процесса.
* Самые важные свойства основного процесса:
  + Измеримость
  + Управляемость
  + Воспроизводимость

Если процесс не обладает этими свойствами – бизнес-цели не могут быть достугнуты.

### Слайд 3. Существующий процесс (схема).

* Описание процесса, описание различных действий по процессу. Основные свойства текущего процесса (не коррелируют с требованиями к процессу, не ведут к достижению бизнес-целей):
  + Не измерим (сколько ресурсов потрачено на то-то и то-то, другие метрики)
  + Трудно управляем
  + Неповторим

### Слайд 4. Недостатки существующего процесса.

* Основные перечислены, но самое важное – неизмерим и трудно управляем, не коррелирует с основными бизнес-целями

### Слайд 5. Следствия недостатков.

* Несоответствие бизнес-целям – значит, их невозможно достичь.

### Слайд 6. Оптимизированный процесс.

* Полностью соответствует бизнес-целям организации, помогает их достичь:
  + Измерим
  + Управляем
  + Воспроизводим

### Слайд 7. Достоинства.

* Устраняет недостатки существующего (улучшает существующий по ряду показателей)
* Соотвтетсвует бизнес-целям
* Для проекта настраивается однократно на старте, причем не программистом, а конфигуратором. Участие программистов минимально, внедрение не задерживает проект (задержка на старте на внедрение с лихвой компенсируется выигрышем времени на всех остальных этапах).

### Слайд 8. Требования к проекту и трудозатраты

* Первоначальная настройка макета: ~2,5 ч/месяца, поддержка актуальности: ~0,5 ч/месяца.

Если будет использоваться и развиваться существующий макет – стоимость минимальна, если настраивать новый – аналогично +- (какой софт, кто будет – компетенции и т.п.)

* Стоимость железа минимальна (парк офисных ПК)
* ТСО: стоимость поддержки + стоимость железа. Софт бесплатен.
* Общие трудозатраты на включение проекта в конвейер: ~30-60 ч/часов однократно + до 1,5% общих трудозатрат, в зависимости от проекта
* Планируемая экономия трудозатрат (экспертная оценка) от включения проекта: 10% общих трудозатрат, в зависимости от проекта

### Слайд 9. Схема реализованного варианта настройки серверов.

* Реализовано в «пилотном» режиме на основе нижеследующих требований
* Требования к ПО общие: мин. стоимость владения, сетевой интерфейс, интеграция с системой оповещения (почта, другое), централизация безопасности (LDAP или другое), простота (порог вхождения), расширяемость, наличие исходников, возможность настройки/поддержки своими силами, железо.
* Требования для VCS – распространенность, наличие четко выраженного «ствола», обязательность коммитов
* Требования к CI – интеграция со специфическими ОС (Astra, MCBC, Эльбрус), интеграция с различными VCS
* Требования к BugTracker – связь с VCS, интеграция с хранилищем артефактов, связь с Тасктрекером, связь с ВИКИ
* Требования к TaskTracker – связь с VCS, визуализация (Гант), связь с Багтрекером, связь с ВИКИ, возможность интеграции с СЭД
* Требования к ВИКИ (База Знаний) – связь с VCS, своя для каждого проекта, связь с Багтрекером, связь с Тасктрекером
* Требования к Хранилищу артефактов – связь с Баг и Таск трекерами, с ВИКИ, простота, общедоступность
* Результат – выбор текущей конфигурации, описание схемы серверов, ЦОД – туда все ложится
* Показ Редмайна, проекта Исток-СПО, других, ВИКИ, хранилища.
* Показ Дженкинса, сборка в реалтайме, новые артефакты

### Прочие вопросы

* Хронометраж по Редмайну:
  + Открыть Редмайн 20-40 сек
  + Логин и в проект 30 се
  + Назначение одной задачи до 1,5 минут (тема, текст, даты начала и окончания, назначено, оценка трудозатрат)
  + Итого: при максимум 16 задачах в день получаем до 30 минут в день
  + Выигрыш намного больше – Исток СПО как пример (трудозатраты, прочее)
* CMMI: система позволит начать переход с текущего уровня 1 на уровень 2 и заложить фундамент под переход на уровень 3.

1. **Начальный уровень (уровень зрелости 1)** – это уровень, на котором, по определению, находится любая компания. На этом уровне разработка ПО ведется более-менее хаотично.
2. **Управляемый уровень (уровень зрелости 2)** – здесь уже появляются политики и процедуры организации процессов, утвержденные на уровне компании. Но в полной мере процессы существуют лишь в рамках отдельных проектов.
3. **Определенный уровень (уровень зрелости 3)** – здесь появляется стандартный процесс на уровне всей компании в целом. Это большой и постоянно пополняющийся набор активов процесса – шаблонов документов, моделей жизненного цикла, программных средств, практик и пр. Любой конкретный процесс получается вырезкой, из этого стандартного.
4. **Управляемый количественно уровень (уровень зрелости 4)** подразумевает появление системы измерений в компании, которые происходят на базе стандартного процесса и позволяют количественно управлять разработкой.
5. **Оптимизирующийся уровень (уровень зрелости 5)** подразумевает постоянное улучшение процессов разработки, как постепенных, пошаговых, так и революционных. При этом данные изменения оказываются не вынужденными, а упреждающими проблемы и трудности. Процесс совершенствуется сам и постоянно – есть, реализованы соответствующие механизмы.