

Инженерия программного обеспечения

Лекция 2

Лектор: доцент

Артамонов Евгений Борисович



Гибкие методологии разработки программного обеспечения

- AGILE-технологии позволяют организовать процесс постепенного приближения к цели проекта путем проведения циклов испытаний с корректировкой последующих, основанных на анализе результатов предыдущих

Гибкие методологии разработки программного обеспечения

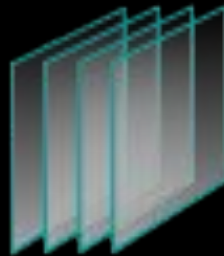


- Scrum (регби схватка) — одна из первых методологий циклического наращивания функциональности и корректировки хода проекта на основе анализа обратной связи от пользователей (первое упоминание 1986, первое описание Швабером и Джефом Сазерлендом [6] на OOPSLA'96).

SCRUM (СК

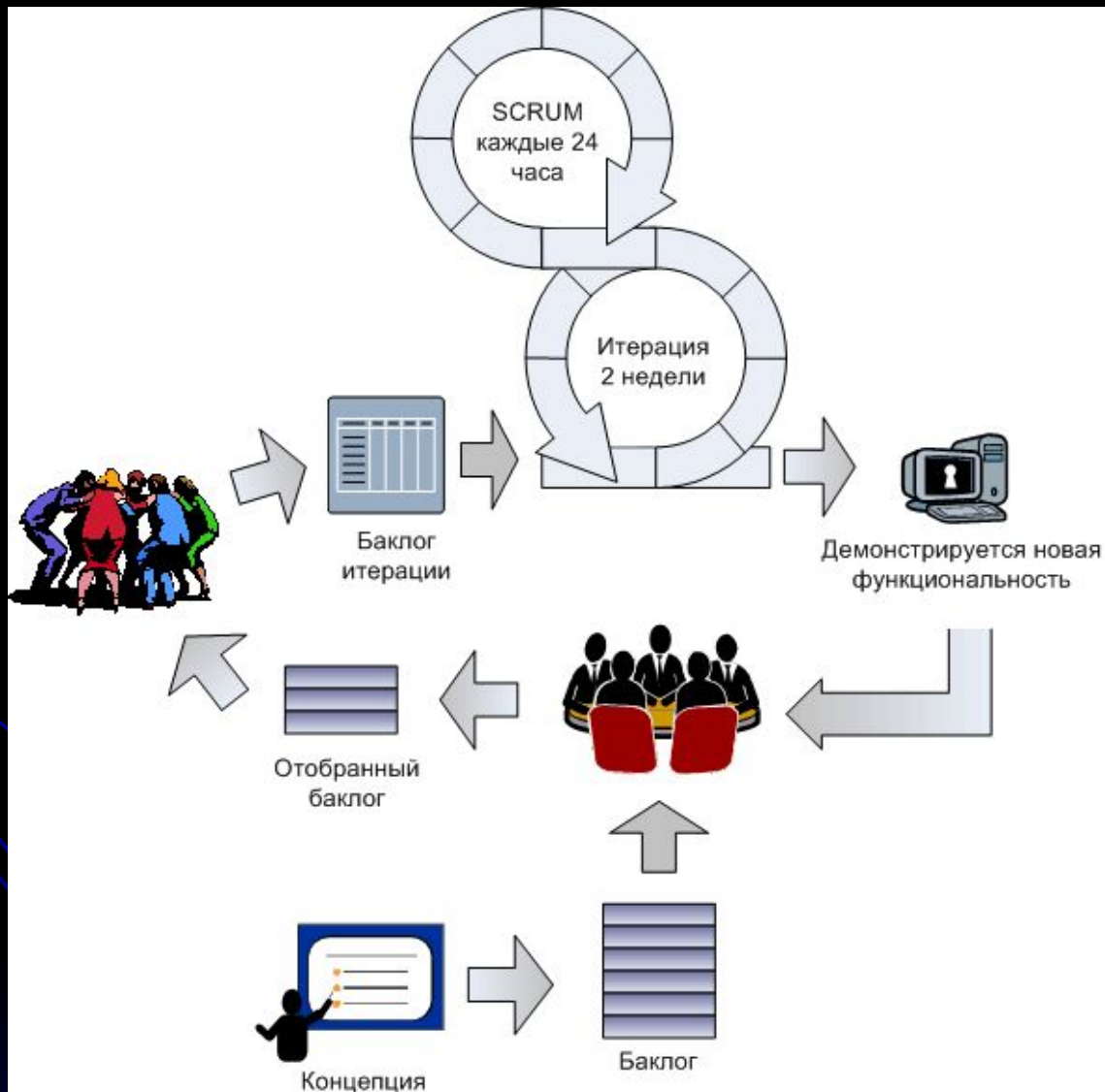
- это набор принципов, на которых строится процесс разработки, позволяющий в жёстко фиксированные и небольшие по времени итерации, называемые спринтами (sprints), предоставлять конечному пользователю работающее ПО с новыми возможностями, для которых определён наибольший приоритет. Возможности ПО к реализации в очередном спринте определяются в начале спринта на этапе планирования и не могут изменяться на всём его протяжении. При этом строго фиксированная небольшая длительность спринта придаёт процессу разработки предсказуемость и гибкость.

SCRUM (СКРАМ)



Резерв проекта

SCRUM (СКРАМ)



Спринт

- итерация в скрам, в ходе которой создаётся функциональный рост программного обеспечения. Жёстко фиксирован по времени. Длительность одного спринта от 2 до 4 недель. В отдельных случаях, к примеру согласно Scrum стандарту Nokia, длительность спринта должна быть не более 6 недель.

Резерв Проекта (Product backlog)

- список требований к функциональности, упорядоченный по их степени важности, подлежащих реализации. Элементы этого списка называются «пожеланиями пользователя» (*user story*) или элементами резерва (*backlog items*).
Резерв проекта открыт для редактирования для всех участников скрам процесса.

Резерв Проекта (Product backlog)

Feature	Estimation, man*it	Priority
Интеграция с IntHR		
Синхронизация сотрудников, орг структуры, позиций и т.д. с учетом multisite	2	Must
ФИН		
Расчет утилизации и проч. финансовых показателей	1	Must
Расчет затрат по реальным зарплатам	1	Must
Редактирование справочных таблиц	1	Must
Закрытый период		
Заказчики		
Внутренние ставки		
Закрытый период по проектам	1	Should
Поддержка разных типов проектов	1	Should
Административный проект подразделения		
Проект обучения		
Multisite		
Синхронизация данных между серверами, управление синхронизацией	4	Must
Отчеты уровня компании	2	Must
Внешний бюджет		
Доработка интерфейса (общий стиль, тулбары, usability, AJAX)	2	Should
Additional view	2	Should
By Month		
By Person		
Прочие доходы	1	Should
Автоматизация PSR	1	Must

Резерв спринта (Sprint backlog)

- **Резерв спринта** — содержит функциональность, выбранную владельцем проекта из резерва проекта. Все функции разбиты по задачам, каждая из которых оценивается скрам-командой. Каждый день команда оценивает объем работы, который нужно проделать для завершения спринта

Резерв спринта (Sprint backlog)

		Days Left in Sprint				
		15	13	10	8	
Who	Description	7/22/2002	7/24/2002	7/26/2002	7/31/2002	
Total Estimated Hours:		554	458	362	270	0
-	User's Guide	-	-	-	-	-
SM	Start on Study Variable chapter first draft	16	16	16	16	
SM	Import chapter first draft	40	24	6	6	
SM	Export chapter first draft	24	24	24	6	
Misc. Small Bugs						
JM	Fix connection leak	40				
JM	Delete queries	8	8			
JM	Delete analysis	8	8			
TG	Fix tear-off messaging bug	8	8			
JM	View pedigree for kindred column in a result set	2	2	2	2	
AM	Derived kindred validation	8				
Environment						
TG	Install CVS	16	16			
TBD	Move code into CVS	40	40	40	40	
TBD	Move to JDK 1.4	8	8	8	8	
Database						
KH	Killing Oracle sessions	8	8	8	8	
KH	Finish 2.206 database patch	8	2			
KH	Make a 2.207 database patch	8	8	8	8	
KH	Figure out why 461 indexes are created	4				

Диаграмма выгорания задач (Burndown chart)

- Диаграмма отображает завершённый спринт. Показывает оставшиеся нерешённые задачи и трудозатраты, необходимые для их завершения из расчёта на 21 рабочий день.
- Диаграмма, показывающая количество сделанной и оставшейся работы.
- Обновляется ежедневно с тем, чтобы в простой форме показать подвижки в работе над спринтом. График должен быть общедоступен.

Диаграмма выгорания задач (Burndown chart)

Sample Burndown Chart

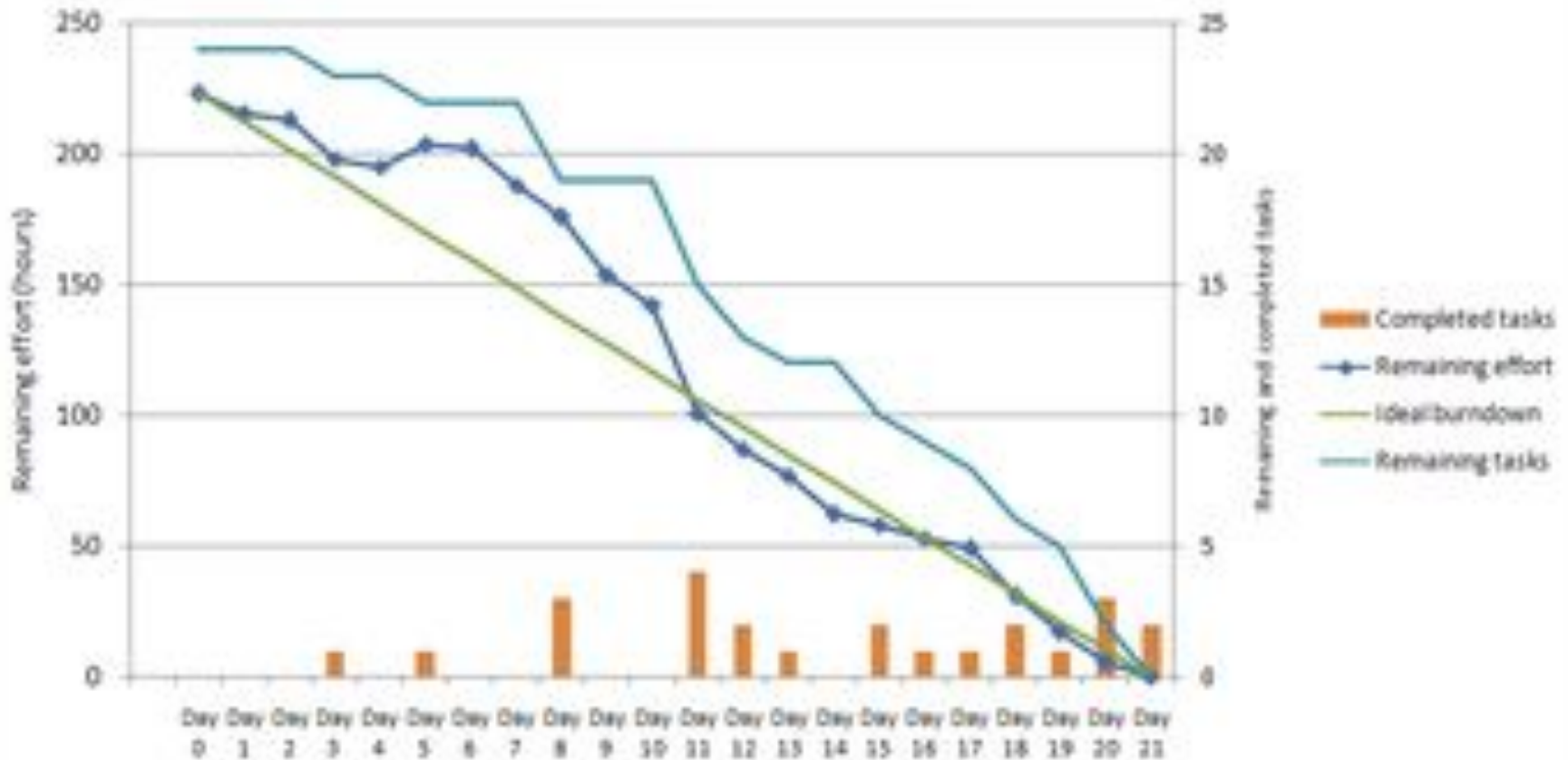


Диаграмма выгорания задач (Burndown chart)

- диаграмма выгорания работ для спринта — показывает, сколько уже задач сделано ("дело выгорело") и сколько ещё остаётся сделать в текущем спринте.
- диаграмма выгорания работ для выпуска проекта — показывает, сколько уже задач сделано и сколько ещё остаётся сделать до выпуска продукта (обычно строится на базе нескольких спринтов).

Дополнительные определения

- История спринта (Sprint Story) - требуемая функциональность, которую добавляют в резерв. Зачастую история имеет следующую структуру: «Будучи пользователем <тип пользователя> я хочу сделать <действие>, чтобы получить <результат>». Такая структура удобна тем, что понятна как разработчикам так и заказчикам.

Дополнительные определения

- Остановка спринта (Abnormal Termination) - может быть произведена раньше срока его планового окончания в исключительных ситуациях. Спринт может остановить команда, если понимает, что не может достичь цели спринта в отведенное время. Спринт может остановить **владелец проекта**, если исчезает необходимость в реализации цели спринта. После остановки спринта проводится совещание с командой, где обсуждаются причины остановки. После этого начинается **новый спринт**.

Дополнительные определения

- Очки за пользовательскую историю - Абстрактная метрика оценки сложности истории, которая не учитывает затраты в человекочасах. Обычно используют одну из следующих шкал: ряд Фибоначчи (1,2,3,5,8,13,21,34,55); линейную шкалу (1,2,3,4 ... n); степень двойки (1,2,4,8 ... 2^n); размеры одежды (XS, S, M, L, XL).

Дополнительные определения

- Задачи истории спринта (Sprint Story Tasks) - Добавляются к историям спринта. Выполнение каждой задачи оценивается в часах. Каждая задача не должна превышать 12 часов (зачастую команда настаивает, чтобы максимальная продолжительность задачи равнялась одному рабочему дню).

Дополнительные определения

- Критерий готовности (Definition of Done (DoD)) - Критерии определяющие степень готовности элемента из *журнала пожеланий пользователя*.
- Скорость команды - Общее количество очков набранных командой за предыдущий спринт. Данная метрика помогает команде понять, сколько историй она может сделать за один спринт.

Роли в скрам-процессе («СВИНЬИ» и «КУРЫ»)

- **Скрам-мастер (ScrumMaster)** — проводит совещания (Scrum meetings) следит за соблюдением всех принципов скрам, разрешает противоречия и защищает команду от отвлекающих факторов. Данная роль не предполагает ничего иного, кроме корректного ведения скрам-процесса. Руководитель проекта скорее относится к **владельцу проекта** и не должен фигурировать в качестве скрам-мастера.
- **владелец проекта (Product Owner)** — представляет интересы конечных пользователей и других заинтересованных в продукте сторон.
- **Скрам-команда (Scrum Team)** — кросс-функциональная команда разработчиков проекта, состоящая из специалистов разных профилей: тестировщиков, архитекторов, аналитиков, программистов и т. д. Размер команды в идеале составляет 7 ± 2 человека. Команда является единственным полностью вовлечённым участником разработки и отвечает за результат как единое целое. Никто кроме команды не может вмешиваться в процесс разработки на протяжении спринта.

Основные обязанности скрам-мастера (менеджер проекта или тимлид)

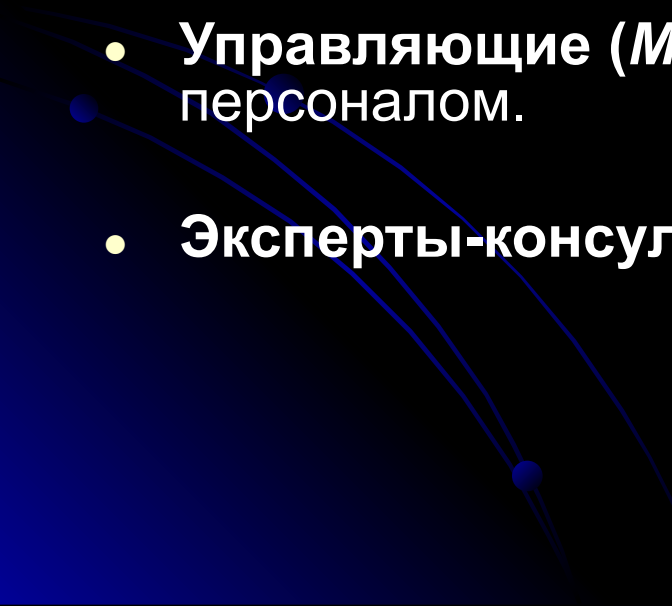
- Создает атмосферу доверия,
- Участвует в митингах в качестве фасилитатора
- Устраняет препятствия
- Делает проблемы и открытые вопросы видимыми
- Отвечает за соблюдение практик и процесса в команде

Основные обязанности product

owner-а (product manager, менеджер проекта или представитель заказчика)

- Отвечает за формирование product vision
- Управляет ROI
- Управляет ожиданиями заказчиков и всех заинтересованных лиц
- Координирует и приоритизирует Product backlog
- Предоставляет понятные и тестируемые требования команде
- Взаимодействует с командой и заказчиком
- Отвечает за приемку кода в конце каждой итерации

Роли в скрам-процессе («свиньи» и «куры»)

- Пользователи (*Users*)
 - Клиенты, Продавцы (*Stakeholders*) — лица, которые инициируют проект и для кого проект будет приносить выгоду. Они вовлечены в скрам только во время **обзорного совещания по спринту** (Sprint Review).
 - Управляющие (*Managers*) — люди, которые управляют персоналом.
 - Эксперты-консультанты (*Consulting Experts*)
- 

Встречи

- Планирование спринта (Sprint Planning Meeting) - происходит в начале новой итерации Спринта
 - Из резерва проекта выбираются задачи, обязательства по выполнению которых за спринт принимает на себя команда;
 - На основе выбранных задач создается резерв спринт. Каждая задача оценивается в идеальных человеко-часах;
 - Решение задачи не должно занимать более 12 часов или одного дня. При необходимости задача разбивается на подзадачи;
 - Обсуждается и определяется, каким образом будет реализован этот объём работ;
 - Продолжительность совещания ограничено сверху 4-8 часами в зависимости от продолжительности итерации, опыта команды и т. п.
 - (первая часть совещания) Участвует **владелец проекта и скрам команда**: выбирают задачи из резерва продукта;
 - (вторая часть совещания) Участвует только команда: обсуждают технические детали реализации, наполняют резерв спринта.

Встречи

- **Ежедневное совещание (Daily Scrum meeting)**

- начинается точно вовремя;
- все могут наблюдать, но только «свиньи» говорят;
- длится не более 15 минут;
- проводится в одном и том же месте в течение спринта.

В течение совещания каждый член команды отвечает на 3 вопроса:

- Что сделано с момента предыдущего ежедневного совещания?
- Что будет сделано с момента текущего совещания до следующего?
- Какие проблемы мешают достижению целей спринта? (Над решением этих проблем работает *скрам мастер*. Обычно это решение проходит за рамками ежедневного совещания и в составе лиц, непосредственно затронутых данным препятствием.)

Встречи

Скрам над скрамом (Scrum of Scrums)

Проводится после ежедневного скрам совещания. Позволяет нескольким скрам командам обсуждать работу, фокусируясь на общих областях и взаимной интеграции. Повестка та же, что и на ежедневном скрам совещании плюс следующие вопросы:

- Что каждая команда сделала с момента предыдущего ежедневного совещания?
- Что каждая команда сделает к следующему ежедневному совещанию
- Есть ли проблемы, мешающие или замедляющие работу каждой команды?
- Нужно ли другой команде сделать что-то из задач вашей команды?

Встречи

Обзор итогов спринта (Sprint review meeting)

Проводится после завершения спринта.

- Команда демонстрирует инкремент функциональности продукта всем заинтересованным лицам.
- Привлекается максимальное количество зрителей.
- Все члены команды участвуют в демонстрации (один человек на демонстрацию или каждый показывает, что сделал за спринт).
- Нельзя демонстрировать незавершенную функциональность.
- Ограничена 4-мя часами в зависимости от продолжительности итерации и инкремента продукта.

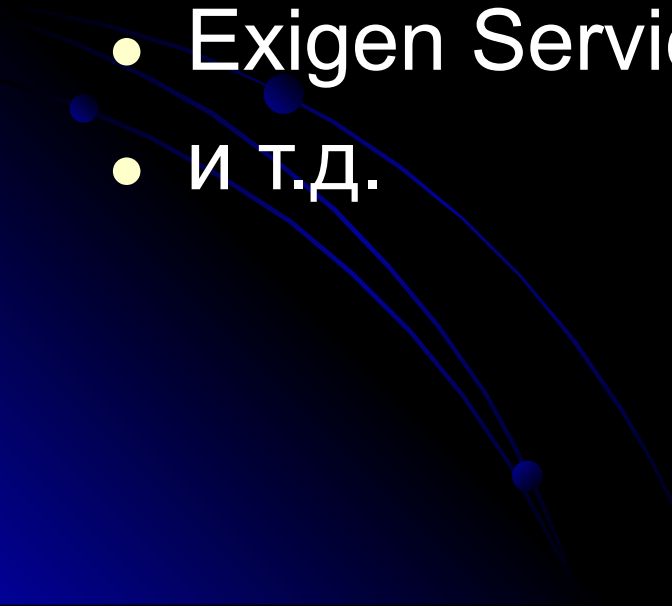
Встречи

Ретроспективное совещание (Retrospective meeting)

Проводится после завершения спринта.

- Члены команды высказывают своё мнение о прошедшем спринте.
- Отвечают на два основных вопроса:
 - Что было сделано хорошо в прошедшем спринте?
 - Что надо улучшить в следующем?
- Выполняют улучшение процесса разработки (решают вопросы и фиксируют удачные решения).
- Ограничена 1—3-мя часами.

Инструментарий для Scrum

- RallyDev,
 - JIRA,
 - AgileTrack,
 - TargetProcess,
 - Exigen Services StarSoft,
 - И т.д.
- 

Домашнее задание

- Разобрать методологию Crystal



Спасибо за внимание!!!
Встретимся на лекции через
неделю

Найти лектора можно в аудитории 5-214

или

по e-mail: eart@ukr.net

Дополнительные материалы на сайте:

eart.ho.ua/prepad.html