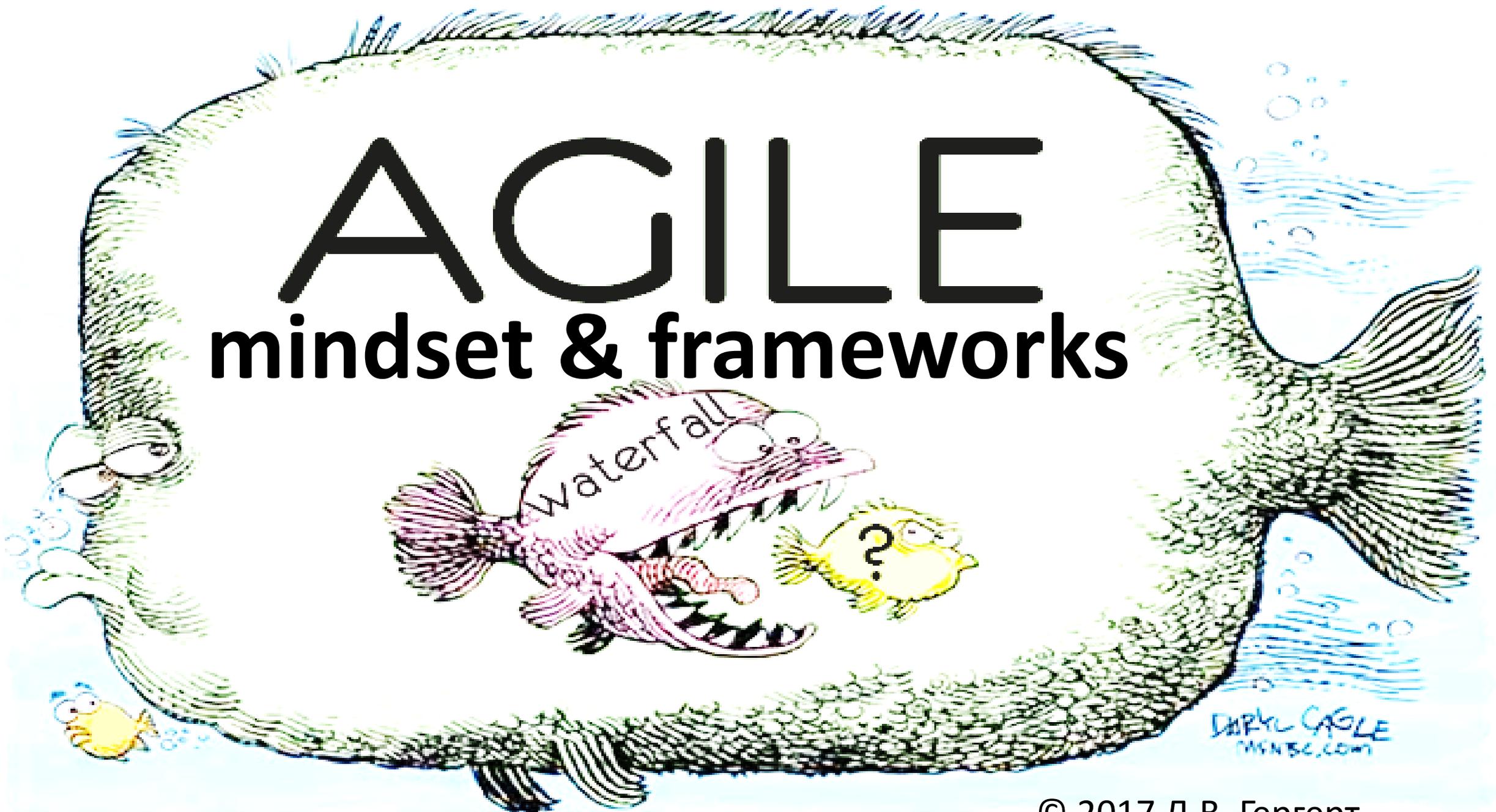


AGILE

mindset & frameworks



**Не освоишь Agile
сегодня, станешь
лузером завтра**

Герман Греф



НЕЛЬЗЯ ПРОСТО ТАК ВЗЯТЬ

И ВНЕДРИТЬ AGILE



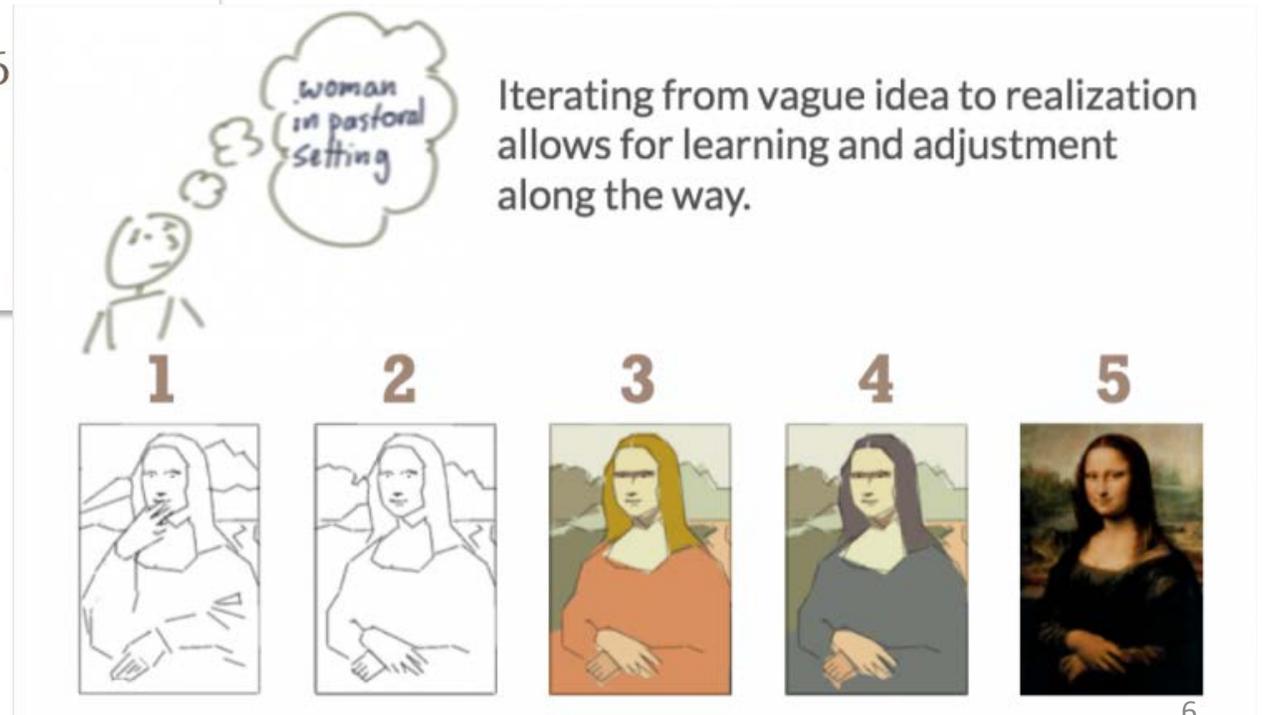
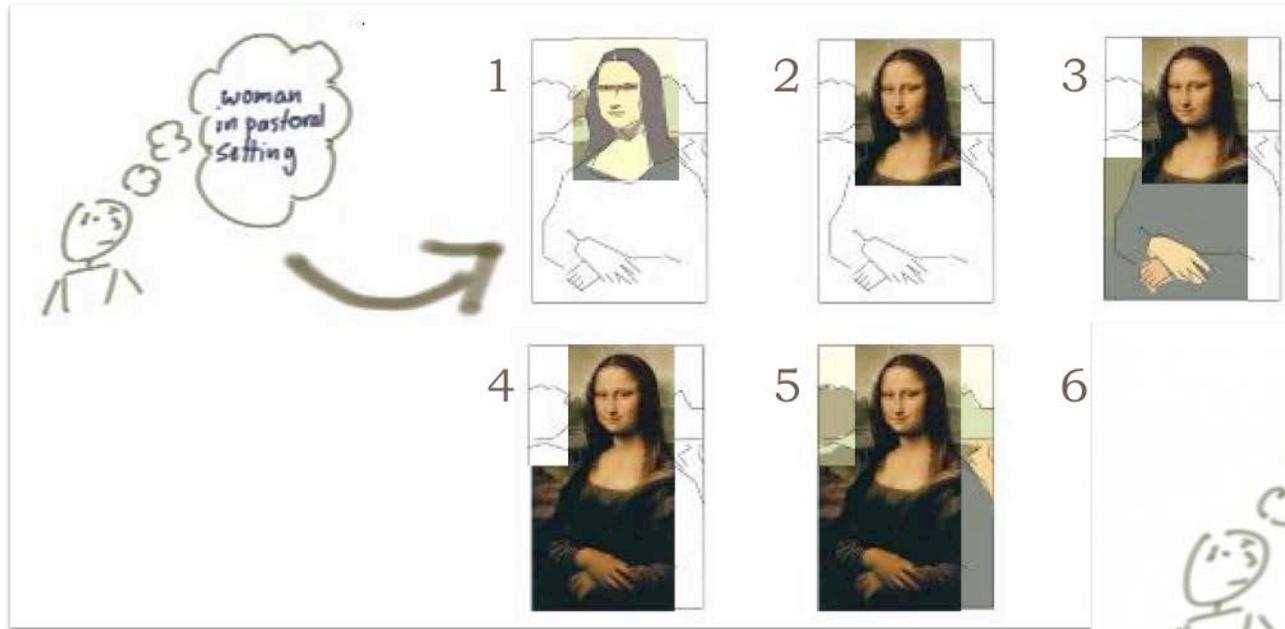
Agile манифест

- **Манифест гибкого подхода к разработке программных продуктов**
- **Ценности:**
 - **Люди и взаимодействие** важнее процессов и инструментов;
 - **Работающий продукт** важнее исчерпывающей документации;
 - **Сотрудничество с заказчиком** важнее согласования условий контракта;
 - **Готовность к изменениям** важнее следования первоначальному плану.
- Таким образом, не отрицая важности того, что справа, всё-таки больше ценится то, что слева.

Agile – это про создание программных продуктов?!

- Про создание продуктов, для которых не существует чёткого и понятного плана «как сделать это правильно».
- Такой план невозможно составить, например, если у вас:
 - продукт, которого ранее не существовало на рынке (не подтверждена ценность идеи и нет достоверной информации по размеру рынка);
 - технологически сложный продукт, который нельзя собрать из набора готовых компонент.
- А может ваши гипотезы о назначении продукта вообще находятся в такой области, где рынок еще не успел сформироваться?

Итеративная и инкрементная разработка



Complexity theory / Теория сложности (запутанности)

- Система это любая сеть, в которой существует некая согласованность. Система может быть довольно неопределенной, у нее может быть, а может и не быть цели.
- Агент это все что угодно, что действует внутри системы. Это может быть человек, группа людей, идея, правила, и тп.

- По материалам «Модель Кеневин (Cynefin framework)» http://www.unusual-concepts.ru/blog/2012/12/cynefin/?&tpwf_mode=main

Виды систем

- 3 вида имеющихся в окружающем нас мире систем:
- **Системы упорядоченные:** в этом случае система ограничивает ее агентов, она основана на упрощениях и правилах, детерминистическая и независимая от наблюдателя. Два вида:
 - Упорядоченные простые
 - Упорядоченные сложные
- **Системы хаотичные:** агенты системы не ограничены и независимы друг от друга.
- **Системы запутанные:** системы слегка ограничивающие своих агентов, где агенты в свою очередь изменяют систему взаимодействуя с ней и друг с другом, они ко-эволюционируют и этот процесс необратим.

Отличия систем

Системы
сложные
(запутанные)

Упорядоченные
сложные

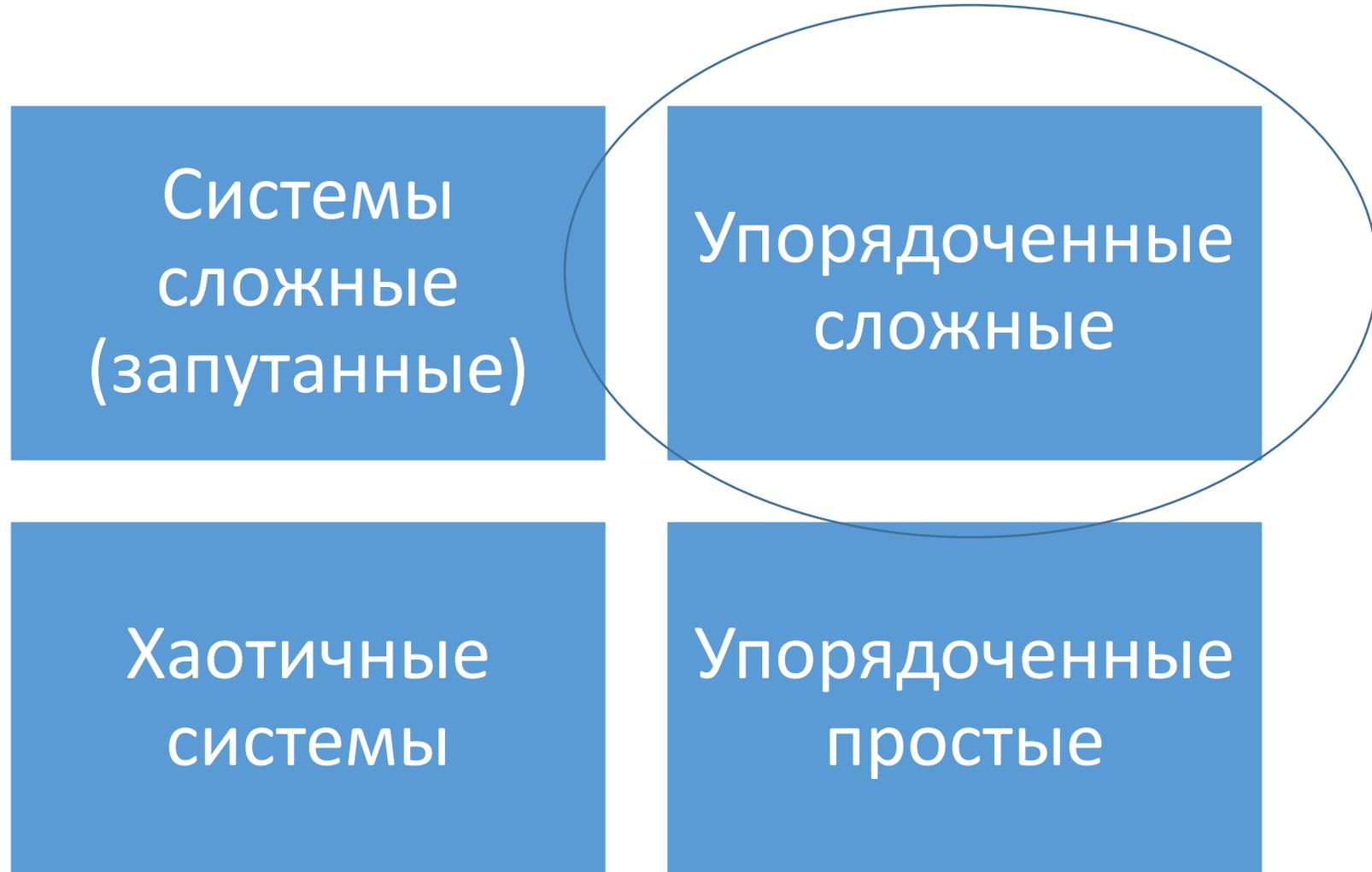
Хаотичные
системы

Упорядоченные
простые

Упорядоченные простые

- Характеризуются своей простотой, взаимосвязи в этих системах очевидны любому разумному человеку, причинно-следственные связи в этих системах тоже ясны и находятся на поверхности.
- Образ действия в подобных системах: **Ощути – Категоризуй – Реагируй**
- В таких системах существует понятие **Наилучшей практики** (Best practice), потому что в них реально найти тот единственно правильный способ достижения результата.

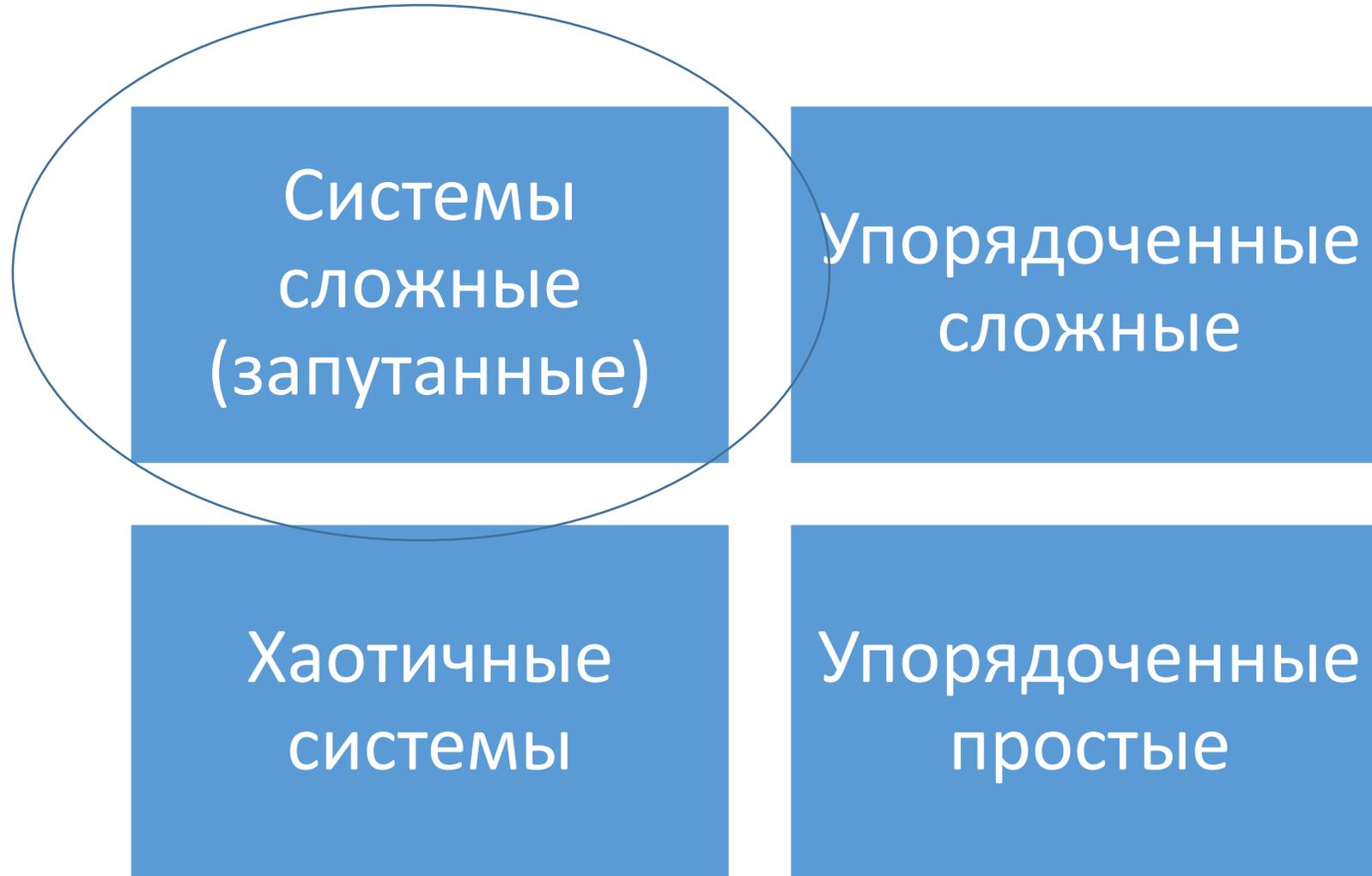
Отличия систем



Упорядоченные сложные

- Система по прежнему упорядочена, причинно-следственные связи есть, но уже не столь очевидны. Это область в которой тон задают эксперты. При наличии необходимых знаний, взаимосвязи можно обнаружить.
- Образ действия в подобных системах: **Ощути – Проанализируй – Реагируй**
- В этой области по прежнему есть правильные ответы, однако их становится несколько. Спросите двух экспертов как достичь того или иного результата и вы получите два совершенно разных ответа, и каждый из экспертов будет доказывать, что его способ наилучший. В этой области пропадает понятие Наилучшей практики, и появляются просто **Хорошие практики** (можешь делать так, можешь по другому, и так и так достигнешь результат).

Отличия систем



Сложные (запутанные)

- Характеризуются настолько запутанными и разнообразными связями между агентами, и содержат такое количество агентов, что в этих системах невозможно не разобраться. Причинно-следственные связи в них не ясны. При двух одинаковых воздействиях на систему результаты могут оказаться совершенно различными. В этой области можно найти достаточное количество фактов, чтобы доказать всевозможные и даже противоречащие друг-другу теории. Минимальное воздействие на систему, может оказать большой эффект.
- Образ действия в подобных системах: **Прозондируй – Ощути – Реагируй**
- В этих системах мы начинаем строить гипотезы и создавать всевозможные эксперименты, чтобы эти гипотезы подтвердить или развеять. Для каждой возникшей вменяемой точки зрения или теории мы создаем эксперимент или серию экспериментов. Эти эксперименты не обязательно должны увенчаться успехом, но должны обеспечить проникновение в суть и добавить понимания происходящего. Эксперименты могут запускаться в параллели и противоречить друг-другу.
- В этой области нет ни наилучшей ни хороших практик. Практики тут возникают по мере проведения экспериментов.

Отличия систем

Системы
сложные
(запутанные)

Упорядоченные
сложные

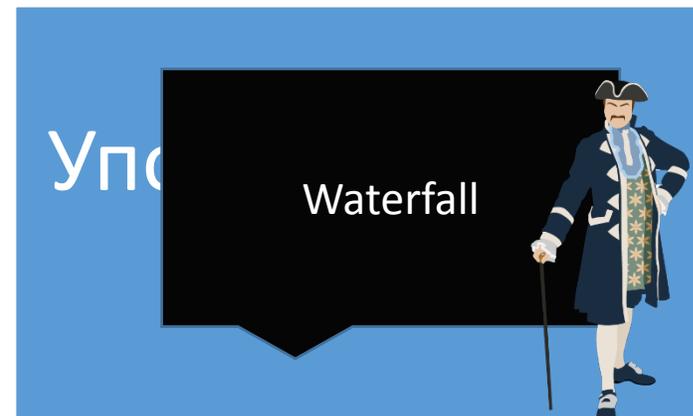
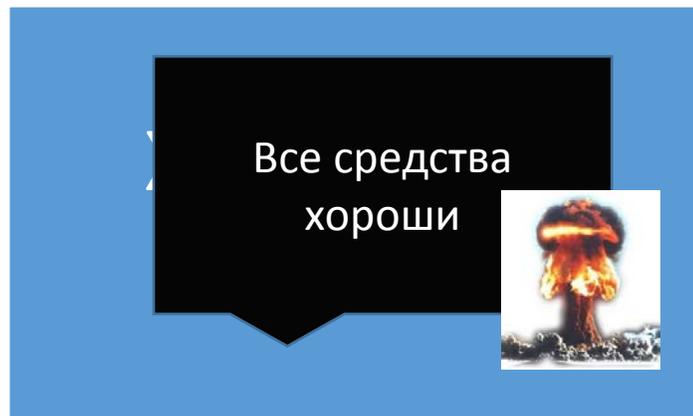
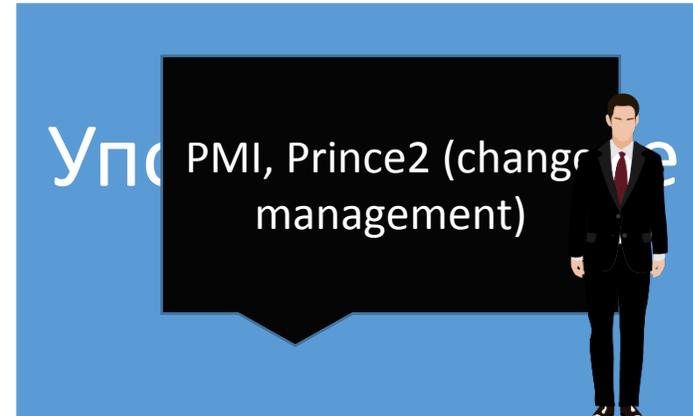
Хаотичные
системы

Упорядоченные
простые

Хаотичные

- Хаос состояние временное. Полный хаос в природе существует очень недолго и лишь в качестве некоего переходного состояния. Причинно-следственных связей нет. Ничего не понятно. Невозможно делать никаких выводов.
- Образ действия в хаосе: **Действуй – Ощути – Реагируй**
- В состоянии хаоса надо действовать и действие должно быть направлено на стабилизацию положения. Пока мы будем оставаться в хаосе, ничего хорошего ждать не придется. Хаос это часто состояние кризиса и/или инноваций. Из состояния хаоса есть два возможных выхода: введение в систему жестких ограничений и ее переход в упорядоченную и простую, либо быстрыми действиями снижающими турбулентность позволить системе перейти в состояние запутанной.
- Практики в состоянии хаоса будут новыми и придуманными впервые.

Методы управления проектами



Frameworks

Change Management

- Среди основных изменений, которые могут существенно повлиять на выполнение проекта, можно выделить следующие:
 - изменения сроков выполнения;
 - изменения стоимостных критериев;
 - изменения методов ведения работ;
 - изменения политики управления;
 - изменения технических условий;
 - изменения потребностей;
 - изменения сроков поставок снабженцами или подрядчиком;
 - изменение ресурсных требований для операций.

Scrum

- Гибкий фреймворк, созданный в 1986 году, считается самым структурированным из семейства Agile. Созданный в 1986 году, он сочетает в себе элементы классического процесса и идеи гибкого подхода к управлению проектами. В итоге получилось очень сбалансированное сочетание гибкости и структурированности.
- Согласно определению, Scrum — это каркас разработки, с использованием которого люди могут решать появляющиеся проблемы, при этом продуктивно и производя продукты высочайшей значимости (с точки зрения клиента).
 - The Scrum Guide. The definitive Guide to Scrum: The Rules of the Game. (Ken Schwaber, Jeff Sutherland)

Элементы SCRUM

Роли

- Владелец продукта
- Скрам-мастер
- Команда

Артефакты

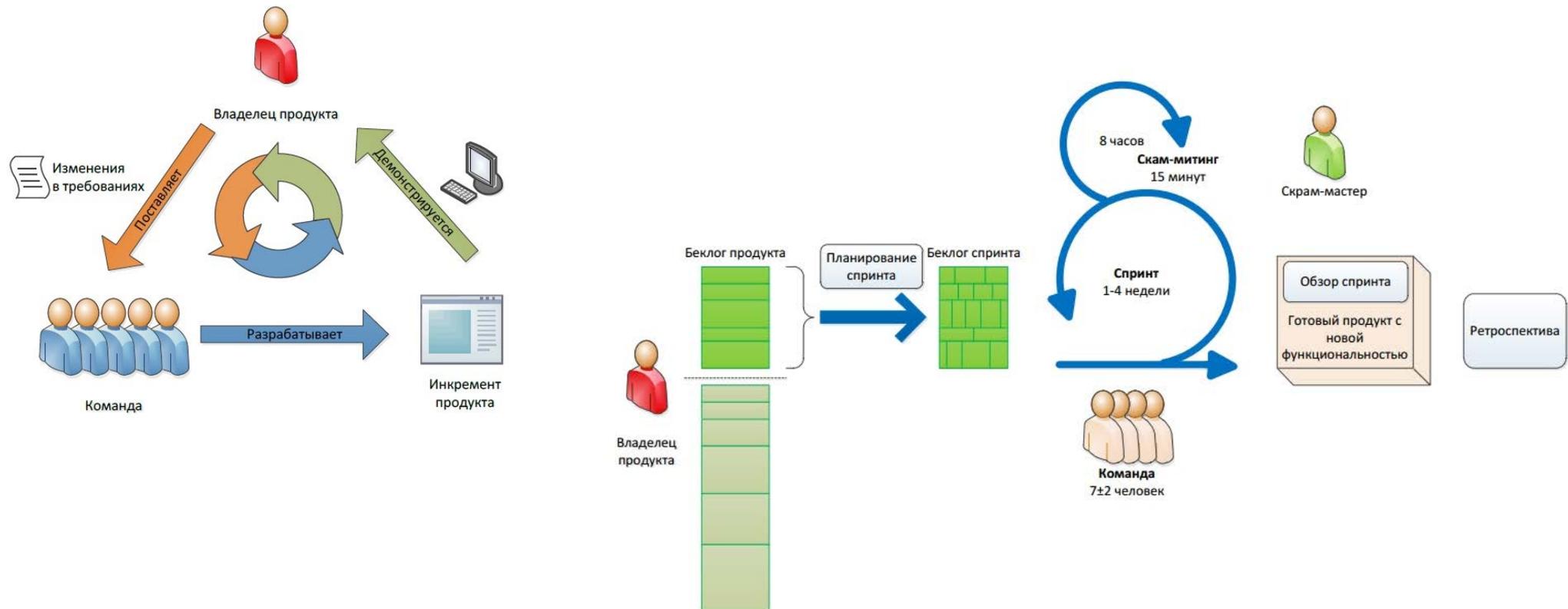
- Беклог продукта
- Беклог спринта
- Инкремент продукта

Процессы

- Планирование спринта
- Обзор спринта
- Ретроспектива
- Ежедневный Скрам

Схема SCRUM

- Скрам использует итеративно-инкрементальный подход



Сильные стороны Scrum

- Scrum был разработан для проектов, в которых необходимы «быстрые победы» в сочетании с толерантностью к изменениям. Кроме того, этот фреймворк подходит для ситуаций, когда не все члены команды имеют достаточный опыт в той сфере, в которой реализуется проект – постоянные коммуникации между членами командами позволяют недостаток опыта или квалификации одних сотрудников за счёт информации и помощи от коллег.
- Онлайн телеканал Netflix является отличным примером быстрых поставок результатов. Сайт ресурса обновляется каждые две недели благодаря Scrum, который не просто позволяет работать с высокой скорости, но и аккумулирует пользовательский опыт и даёт возможность выявить самое главное для клиентов.
- В ходе каждой итерации, разработчики добавляют и тестируют новые функции сайта и убирают те, которыми не пользовались клиенты. По словам команды Netflix, основное преимущество Scrum в том, что он позволяет «быстро ошибаться». Вместо того, чтобы долго и с большими затратами готовить крупный релиз, поставки раз в две недели по Scrum имеют небольшой размер. Их легко отслеживать и, если что-то идёт не так, быстро исправлять.

Слабые стороны Scrum

- Scrum очень требователен к команде проекта. Она должна быть небольшой (5-9 человек) и кроссфункциональной – то есть члены команды должны обладать более чем одной компетенцией, необходимой для реализации проекта. Например разработчик ПО должен обладать познаниями в тестировании и бизнес-аналитике. Делается это для того, чтобы часть команды не «простаивала» на разных этапах проекта, а также для того, чтобы сотрудники могли помогать и подменять друг друга.
- Кроме того, члены команды должны быть «командными игроками», активно брать на себя ответственность и уметь самоорганизовываться. Подобрать такую зрелую команду очень непросто.
- Scrum подходит не для всех команд и организаций ещё и потому, что предлагаемый процесс может не подойти для разработки конкретного продукта – например промышленного станка или постройки здания.

Спасибо за внимание