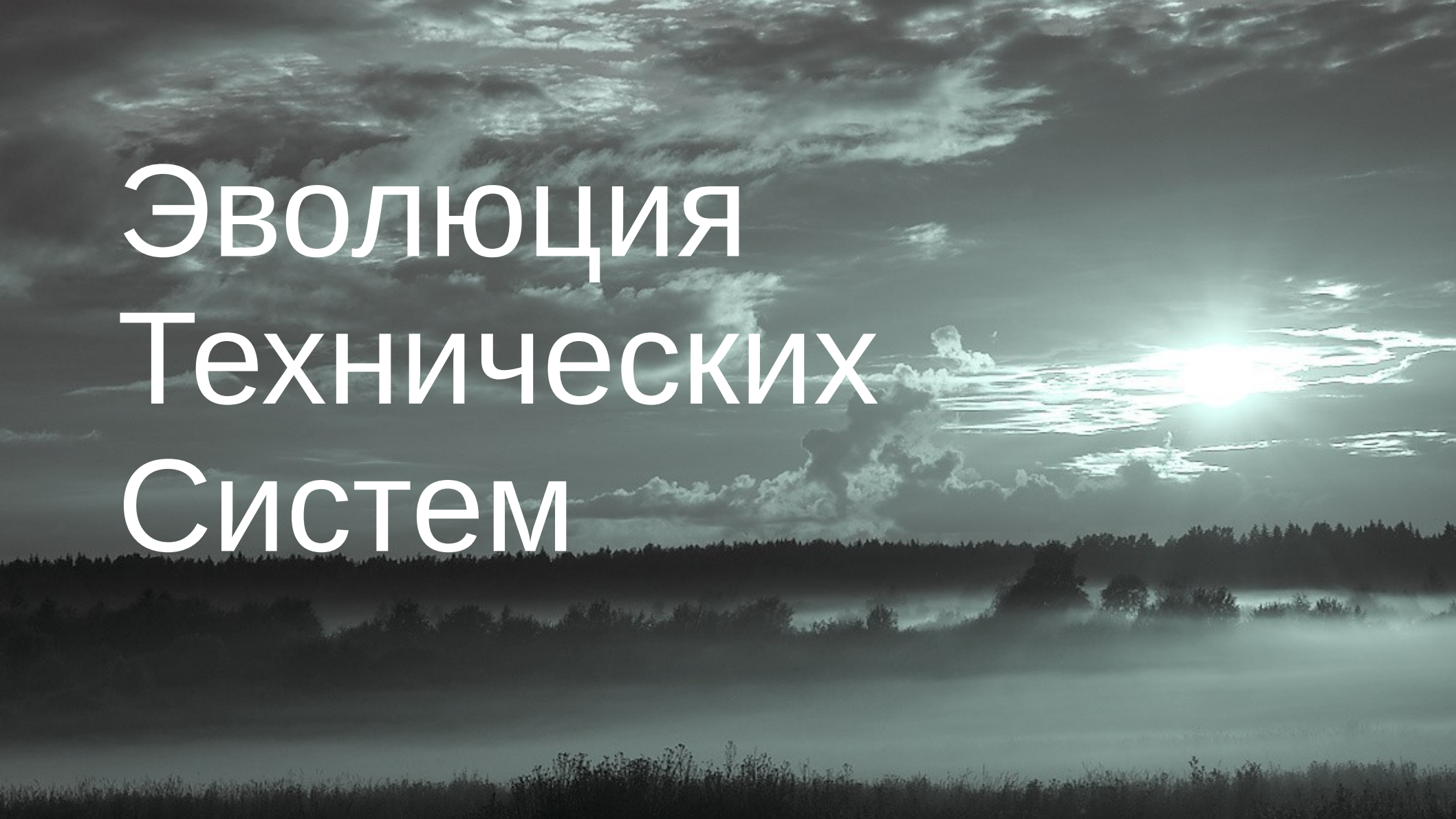


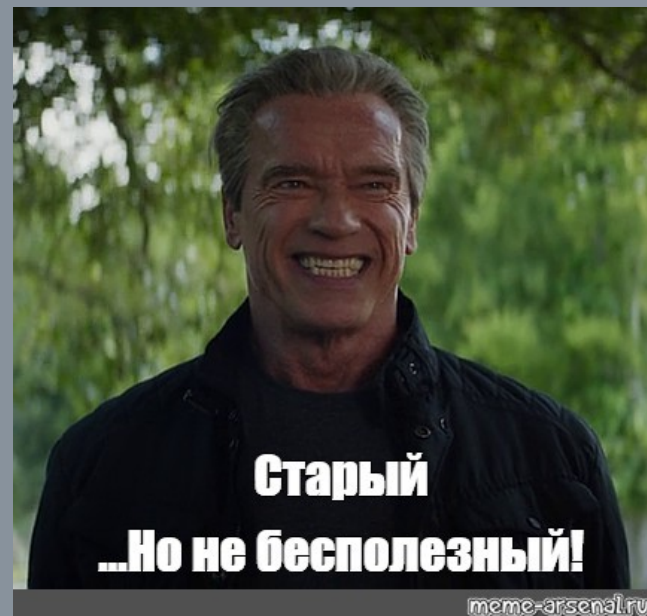
Эволюция Технических Систем



О чём поговорим?

Давайте поговорим о технических системах,
от неолитического ретро до острия
прогресса.

И рассмотрим понятие
устаревания.



Что общего у этих объектов?



Они появились очень давно и дожили до 2025

- Колесо — 5000 лет до н.э.
- Папоротник — 400 млн. лет назад (девон)

Что с устареванием?

Можем ли мы считать, что колесо устарело?

- Базовая идея и функционал одинаковые
- Технологии
улучшились
ЭВОЛЮЦИОННО



Ещё примеры старья

Автосцепка СА-3

- Разработана в 1932 году в СССР
- Основа —
система Виллисона
(1910 год)



Ещё примеры старья

Знакомая всем устаревшая система.

Убивает уверенно.

Автомат Калашникова
1947 год



Ещё примеры старья

Пулемёт «Максим»

- 1897 год: Хайрем Максим подстригает лужайку
- 2025 год — конфликт на Украине



Ещё примеры старья

УАЗ СГР «буханка», 469

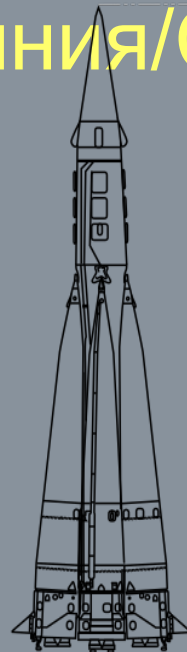
- 1965 год
- Никогда не будут сняты с вооружения
- Простая конструкция, живучесть и ремонтпригодность, нетребовательность к топливу



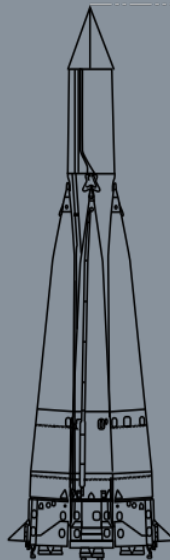
Ещё примеры старья

Ракета Р-7/Спутник/Восток/
/Восход/Молния/Союз

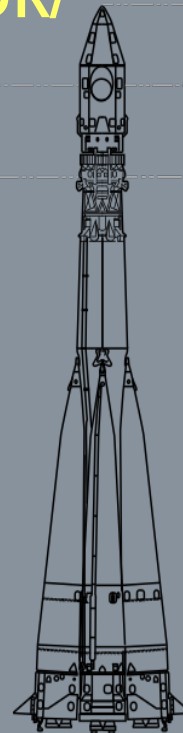
- 1957 год



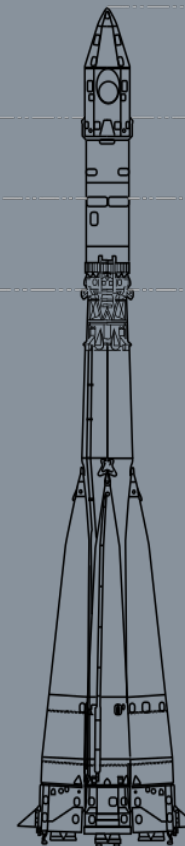
R-7 (8K71)
Test vehicle
1957



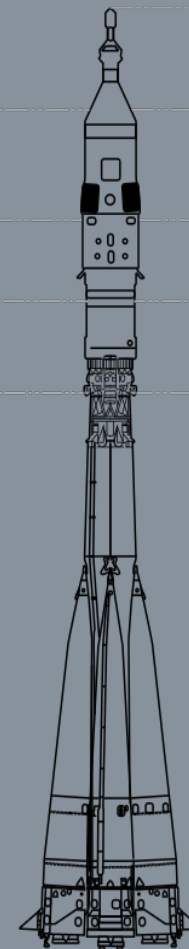
8K71PS
Sputnik (PS) Launcher
1957



8K72K
Vostok (3KA) Launcher
1960



11A57
Voskhod (3KV) Launcher
1963



11A511
Soyuz (7K-OK) Launcher
1966

49,3 m

44,418 m

38,36 m

34,22 m

29,167 m

Ещё примеры старья — в ИТ

- IBM/360 и потомки (z/Series): 1964
- Intel 8086 и потомки (AMD64): 1979
- Intel 8051: 1980
- ARM: 1983-1987
- IBM PC: 1981

Ещё примеры старья — в ИТ

- IEEE RS-232 1960 год
- ASCII: ANSI 1963 год
- TCP/IP: 1970-е года
- IEEE 802.3 Ethernet: 1973-1983
- RFC: IETF, 1969..наши дни

Ещё примеры старья — в ИТ

- COBOL (1960)
- FORTRAN (1957)
- Си (1969)
- UNIX (1970)
- Windows (1985)

К чему ты клонишь?

А давайте создадим модель :)

- Для предсказания
жизненности системы
- Для объяснения
парадоксов успеха
систем-гадких утят или
провала платиновых систем
- Для осмысленного проектирования систем



Эволюция технических систем

Есть работы Ричарда Докинса, развитые Дэвидом Дойчем. В них делается обобщение эволюции на любые информационные процессы во Вселенной.

- Давайте рассматривать технические системы как биологические виды

Эволюция технических систем

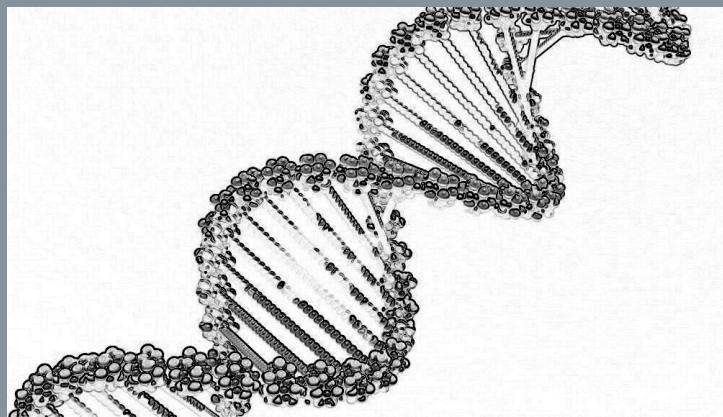
База нео-эволюционной теории:

- Мем (ген — частный случай мема)
- Репликатор
- Мемы реплицируются репликаторами
- Мемы соперничают
- Не умер — победил!
- Более сильный репликатор вытесняет соседние виды в нише

Мемы

Мемы — это базовая структурная единица информации

- База биологических систем — это ген
- База технических систем — это проект



Мемы

- Мемы/гены являются концентрированным знанием об окружающей среде
- Мемы/гены либо соответствуют среде, либо уничтожаются средой

Мемы

Свойства мемов:

- Меметичность (мощность репликатора)
- Программа (что он делает?)
- Устойчивость к помехам (автокоррекция)

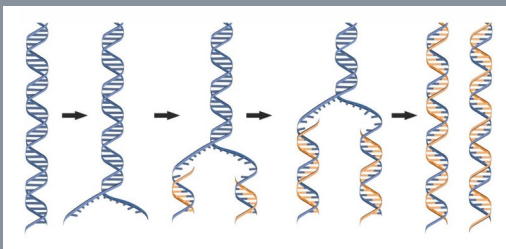
Мемы

Условие выживания мема:

- Эксплуатация потребности

Кстати, это необязательно.

Мем может менять поведение носителя, и форсировать потребность в себе (как паразит)



Репликаторы

Что является репликатором для тех.систем?

- Стандарты
- Совершенство системы
- Пиратство
- Реклама (инфоцыг)



По отдельности и вместе позволяет нарастить пользовательскую базу до порога самовоспроизводства.

Стандарты

ЕСКД — комплекс стандартов, устанавливающих взаимосвязанные нормы и правила по разработке, оформлению и обращению конструкторской документации, разрабатываемой и применяемой на всех стадиях жизненного цикла изделия (при проектировании, изготовлении, эксплуатации, ремонте и др.)

Основное назначение ЕСКД — установление единых оптимальных правил выполнения, оформления и обращения конструкторской документации.

Структура обозначения номера стандарта ЕСКД

ГОСТ 2.301-68

- Последние две цифры года регистрации стандарта (1968)
- Номер стандарта в классификационной группе
- Номер классификационной группы
- Цифровой код ЕСКД
- Государственный стандарт

2

Властью создавать стандарты обладают государства и крупные корпорации

- Стандарты являются составляющей суверенитета
- Нужен рынок
- Нужна способность принудить



Стандарты



- Войны способствовали стандартизации (Наполеон, мировые войны, итд)
- Рождаются в противостоянии, а не в кооперации (иначе была бы полная совместимость всего со всем и полная ремонтпригодность)
- Цель стандарта — в контроле рынка

Стандарты



Примеры форсед мемов от Государства

- Платёжная система «Мир» (помните сколько было сарказма? Не помните?)
- Госуслуги (помните видео с Джигурдой и ботаном? Какие на#\$% госуслуги, вали отсюда ботаник херов! А ты! А ты — чубакка б%:ть!)

Принуждалка — отросла!



Стандарты

А кто их создаёт?

- А создают их внутри корпораций и внедряют на рынки в виде массовых продуктов
- Vendor Lock — это корпоративный стандарт
- Все оказываются перед фактом — вот это теперь стандарт. Можете им не пользоваться, тогда вы вне экосистемы, лузер и лишенец
- Экосистема Apple: не стандарт, но стандарт

Стандарты

В каком виде они появляются на бумаге изначально?
А по-разному:

- Меморандумы (RFC-alike)
- Патенты
- Монографии и фундаментальные работы
- Чертежи, спецификации и техническая документация систем

Стандарты

Что первично, система или стандарт?

- Опыт показывает что скорее система
- Стандартам в ИТ мало кто следует на 100%, они исключительно сложны, многословны и нереализуемы в полной мере

Стандарты

Кто кодифицирует стандарты?

- Государственные институты стандартизации
- Международные институты стандартизации (что вообще означает «международные»?)
- Консорциумы
- Комитеты по стандартизации

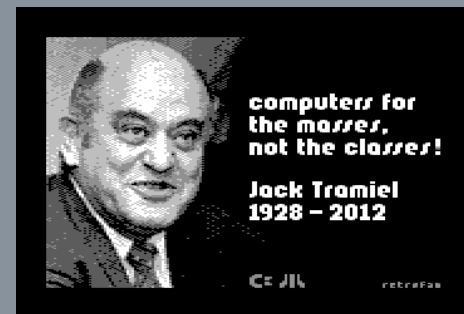
Стандарты

Итог:

- Это мощный репликатор, бустит системы только так
- Заставят пользоваться, даже если не хочешь
- Не только лишь все могут вводить стандарты, чеканить монету и судить суды

Совершенство систем

Если мы хотим захватить рынок, но мы всего лишь маленький человечек...



(второй справа дед — Федерико Фаджин (Zilog Z80))

Совершенство систем

...тогда мы должны создать такую систему, которая сама пробьёт путь к пользаку. Эта система по своим свойствам должна быть сравнима с травой, или с борщевиком (без его ядовитых свойств) — неприхотливой, пробивной, и в то же время закрывающей большинство потребностей...

Совершенство систем

Для этого система должна быть:

- Простой
- Дешёвой

Совершенство систем

Простота:

- Второй закон термодинамики системно понижает сложность всего-превсего
- Сложные системы противоречат физике нашей Вселенной
- «Сложность, порождаемая из простоты» является одним из определений красоты
- Мы знаем это как «не умножай сущностей» или бритву Оккама

Совершенство систем

Чтобы выжить, система НЕ обязана исполнять своё предназначение абсолютно идеально, на порядки лучше конкурентов, быть устроена оптимально и быть сверх совершенной.

Чтобы выжить и победить, Система может быть устроена чудовищно неэффективно, некрасиво и неизящно, если при этом она чуть лучше остальных.

Совершенство систем

Подходы к совершенству:

- Наличие абсолютно необходимого
- Отсутствие лишнего:
 - Примитивность на грани одноразовости
- Задел расширяемости:
 - Открытые архитектуры

Совершенство систем

Простую систему просто изменять.

Простую систему просто понять.

Простую систему просто скопировать.

Пиратство как репликатор

- Пиратство это репликатор :)
- Является негласной стратегией многих корпораций
- Некоторые корпорации активно борются с пиратством (Sony)
- Пиратство позволяет получить преимущество слабой, несовершенной системе, над более совершенной, которую не пиратят

Цикл жизни системы

Родовые травмы:

- Самое опасное время для системы — первые недели и месяцы внедрения
- Реальность выявляет все недочёты и огрехи
- Если система выжила, первая опасность миновала

Цикл жизни системы

Эффект Линди

- Если система прожила N лет, она проживёт ещё N лет
- Прожитый срок — показатель устойчивости, жизнестойкости, полезности, качества

Это прикол актёров ТВ, они собирались в Lindy's delicatessen и шутили, что если не уволили после первого сезона, доработаешь до 70 лет в сериале)

r/K-стратегии

- r-стратегия: огромное потомство, огромная смертность потомства, короткое время жизни, выживают единицы, низкая адаптивность, быстрая смена поколений, быстрое распространение. Дешёвое потомство.
- K-стратегия: малое потомство, высокое время жизни, высокая адаптивность, низкая смертность. Дорогое потомство.

r/K-стратегии

- r-отбор: оппортунисты. Хорошо для высокоизменчивых катастрофических окружений
- K-отбор: конкуренты. Хорошо для стабильных окружений

r/K-стратегии

$$\frac{dN}{dt} = rN \left(1 - \frac{N}{K} \right)$$

N — численность (или плотность численности) популяции

dN / dt — текущая скорость её роста

r — предельная скорость её роста (коэффициент размножения)

K — переносимый объём (предельная численность или плотность численности популяции, при которой популяция ещё может существовать в равновесии с биотой)

r/K-стратегии

- Agile — типичный r-оппортунист. Здесь и сейчас. Проблемы завтрашнего дня это не наши проблемы. Из миллиона проектов в долгосроке выживет один.
- Waterfall — типичный K-конкурент. Долгие роды. Высокая заложенная адаптивность. Длительная жизнь.

Что особенного в ИТ?

- Закон Мура: очень быстро обновляются возможности
- Софт: работа с чистыми идеями (никакой физики)
- Итог: очень быстрый эволюционный цикл по сравнению с другими отраслями
- Аналог: эволюция в микробиологии

Инфоцыганство как репликатор

Реклама - это последняя надежда проклятых

Дуглас Крокфорд

Продажники, маркетологи, коучи, рекламщики эксплуатируют эффект новизны технологии/продукта. Они усиливают собой репликатор новых продуктов, вне зависимости от их качеств. Важно лишь то, успевает ли продукт набрать пользовательскую базу для самовоспроизводства до того, как эта поддержка будет снята.

Инфоцыганство как репликатор

Серебряные пули:

- ООП
- Визуальное программирование
- CASE-системы

Инфоцыганство как репликатор

Серебряные пули:

- ASP.Net
- CMS
- XML
- Web Services (SOAP, WSDL)
- Silverlight, Adobe Flash
- Системы автоматизации тестирования

Инфоцыганство как репликатор

Серебряные пули:

- REST API
- NoSQL
- Микросервисы
- Блокчейн
- Тысячи их!

Инфоцыганство как репликатор

От коллеги из клуба:

...Джоель Спольски, которого я в нулевых активно почитывал (и почитал) называл это стрельбой для прикрытия. Суть его мысли в том, что крупные копропации (вы конечно поняли о ком идет речь, правда?) вбрасывают "новые технологии", убеждая более мелкие бросать все дела и переписывать свой код по-новому, "ибо это и есть истинный путь настоящих джедаев, говорю вам я!", а покуда они в сотый раз переделывают выдают на-гора новые фишки, написанные исключительно с использованием своего собственного дремучего Легаси...

Инфоцыганство как репликатор

Серебряные пули: итог

- Технологии перестают быть серебряными пулями после снятия маржи с волны продаж
- Технологии просто показывают себя как есть и выживают или нет

Запланированное устаревание

- 1932 год: памфлет Бернарда Лондона «Конец депрессии через планируемое устаревание»
- 1954 год: доклад инженера Брукса Стивенса на рекламной конференции в Миннеаполисе
- 1950-1960-е года: Харви Эрл, дизайнер великой американской автомобильной классики
- Весь XX век — индустрия моды
- 2000-е: Android, Windows, Apple
- 2025: мы находимся здесь
- https://ru.wikipedia.org/wiki/Планируемое_устаревание



Запланированное устаревание

- Это важный фактор, с которым нужно считаться
- Это искусственный фактор
- Инфоцыганство часть стратегии планируемого устаревания
- Хорошие системы в общем преодолевают запланированную смерть
- Длительный срок жизни систем — это норма

Запланированное устаревание

- Норма для коммерческого софта
- Photoshop тот же что в 1997, MS Office тот же что в 1997
- Открытый софт не так подвержен этому — релизы ради релизов не нужны, вдобавок обычно отстаёт по фичам

Анализ: методика

Нужно рассмотреть

- Мем как программу (что делает)
- Эксплуатацию потребности (кому это нужно)
- Репликатор (как размножается)
- Стратегию размножения (r/K)

Анализ: методика

Репликатор:

- Кем форсируется? (государство, вендор)
- Формирует экосистему?
- Пользовательская база обеспечивает воспроизводство?
- Пиратят?

Анализ: методика

Репликатор:

- Открытая система?
- Открытые стандарты?
- Технически совершенна?
- Проста?

Анализ: методика

- Стратегия размножения: r/K? Agile? Waterfall?
- Есть требования?
- Есть ТЗ?
- Есть документация?

Анализ: методика

- Есть вариации вследствие копирования?
Значит началось самовоспроизводство
- Наличие конкурентов с более мощным репликатором

Анализ: методика

Эксплуатация потребностей

- Есть ли объективная потребность в системе?
- Система меняет поведение пользователя в свою пользу?

Анализ: пример

Наши любимые 8-битки, гадкие утята 80-х

- ZX Spectrum
- Commodore C64

Анализ: пример

- Были созданы в небольших компаниях
- Нет силового форсирования
- Простой дизайн
- Простое копирование (особенно Spectrum: 200 вариантов в СССР)

Анализ: пример

ZX Spectrum, 1990-е годы:

- Несколько повторных реализаций и систем по мотивам
- ZX Spectrum, 2020-е годы:
- Куча эмуляторов
- Реализации на FPGA, на одноплатниках и микроконтроллерах

Анализ: пример

Вот коммент от ретроделов-спектрумистов:

это пока.. создатели спектрум некст вот столкнулись с тем, что старая frga больше недоступна к заказу - пришлось переделывать на новую. в итоге - две ревизии, имеющие разные возможности и не полную совместимость

этот эпизод называется "мутация")

Анализ: пример

Prusa i3 RepRap – привет из 2012

- Стоил около 15.000 руб.
- Поставлялся в полностью разобранном виде
- Корпус собирается из плоских деталей, вырезанных из оргстекла
- Держатели шаговых двигателей напечатаны на таком же принтере
- Печатающая поверхность – обычное стекло. Крепится к нагревателю канцелярскими прищепками
- Приводной ремень отмеряешь, отрезаешь и натягиваешь сам
- Прошиваешь через Arduino Studio

Анализ: пример

- RepRap — это самокопирующийся автомат фон Неймана
- Создан как примитивная и готовая к копированию система
- <https://slideplayer.com/slide/16925115/>

Von Neumann's Automaton and Viruses

Most slides taken from
Weizmann Institute of Science and
Rensselaer Polytechnic Institute

Модель

Она вот такая:

- Системы подчиняются законам эволюции и меметики
- Потребность в системе поддерживает либо среда, либо она сама
- Репликаторами для них являются админ.ресурс вендоров или государств, а также внутреннее совершенство, простота (в т.ч. копирования) и открытые интерфейсы
- Критично набрать базу пользователей для самовоспроизводства, а также выдержать первоначальную агрессию среды
- ДЛИТЕЛЬНЫЙ СРОК ЖИЗНИ СИСТЕМЫ — ЭТО НОРМА!!!
Делайте системы на века, увековечьте себя!)



ВСЁ!