

Практики жизненного цикла систем машинного обучения



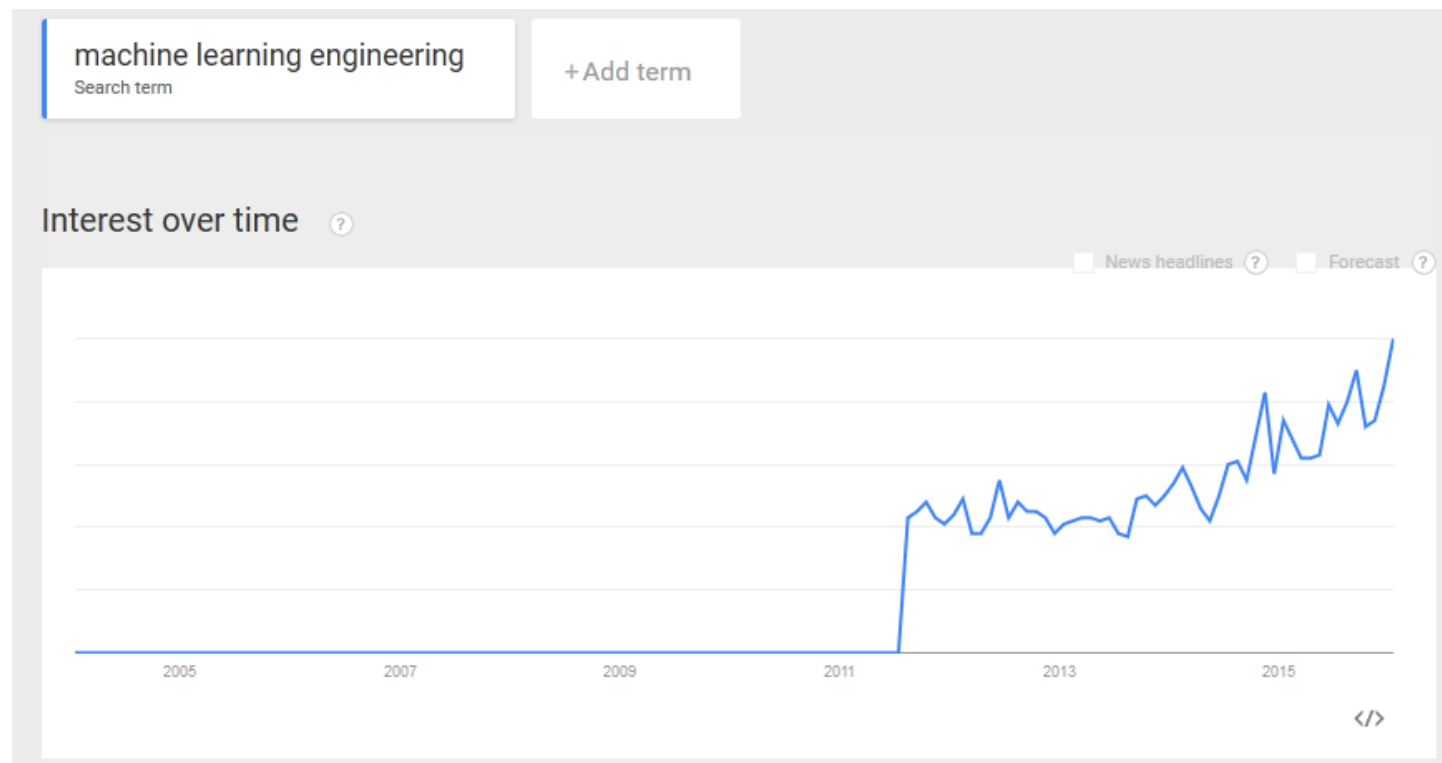
SECR'16
28 октября 2016г.

Kind of *Engineerings*

- Mechanical engineering
- Agriculture engineering
- Aerospace engineering – aircraft architecture
- Systems engineering
- System of systems engineering
- ...
- Software engineering
- Control [systems] engineering – control [system] architecture
- Knowledge engineering -- architecture
- **Machine learning [system] engineering (-- architecture?)**
- ...
- Neural engineering
- neural network engineering -- neural [network] architecture
- **Feature engineering -- ???**

Systems, Software, Machine Learning Engineerings

- Software engineering [term appeared in 1965, boosted by NATO as a profession in 1968]
- Systems engineering [Bell Labs in 1940s, boosted as a profession by NCOSE 1990]
- **Machine learning engineering [term appeared in 2011]**



Вместо «программа» подставьте «система машинного обучения».

Why is my program
not working?

You need to know
why?

You need to
program working
properly?

To repair compiler?

To advance theory?

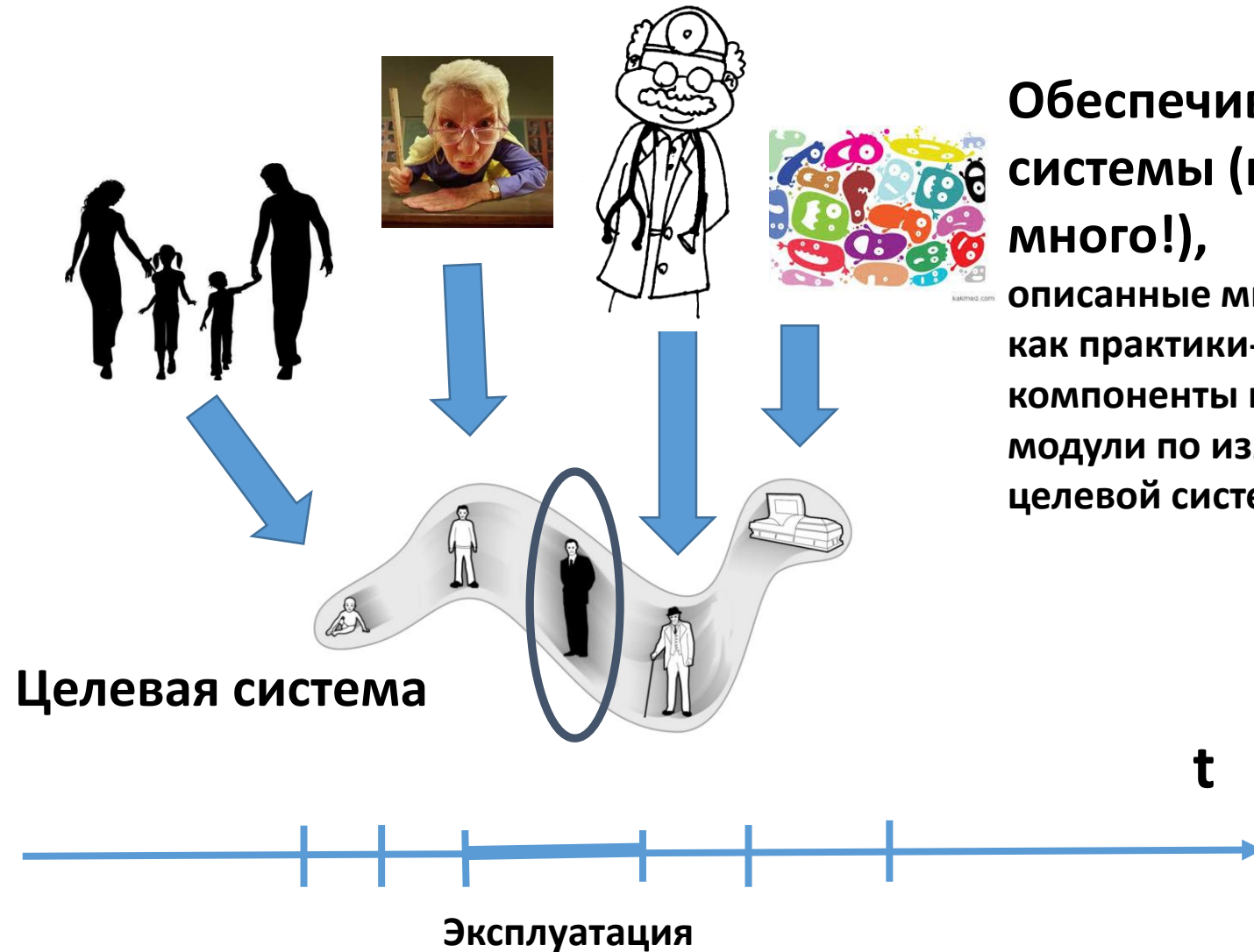
**Software engineer
(application)**

**Software engineer
(systems)**

Computer Scientist

Жизненный цикл системы – это про тех, кто её делает

«Практики
жизненного
цикла
систем
машинного
обучения»

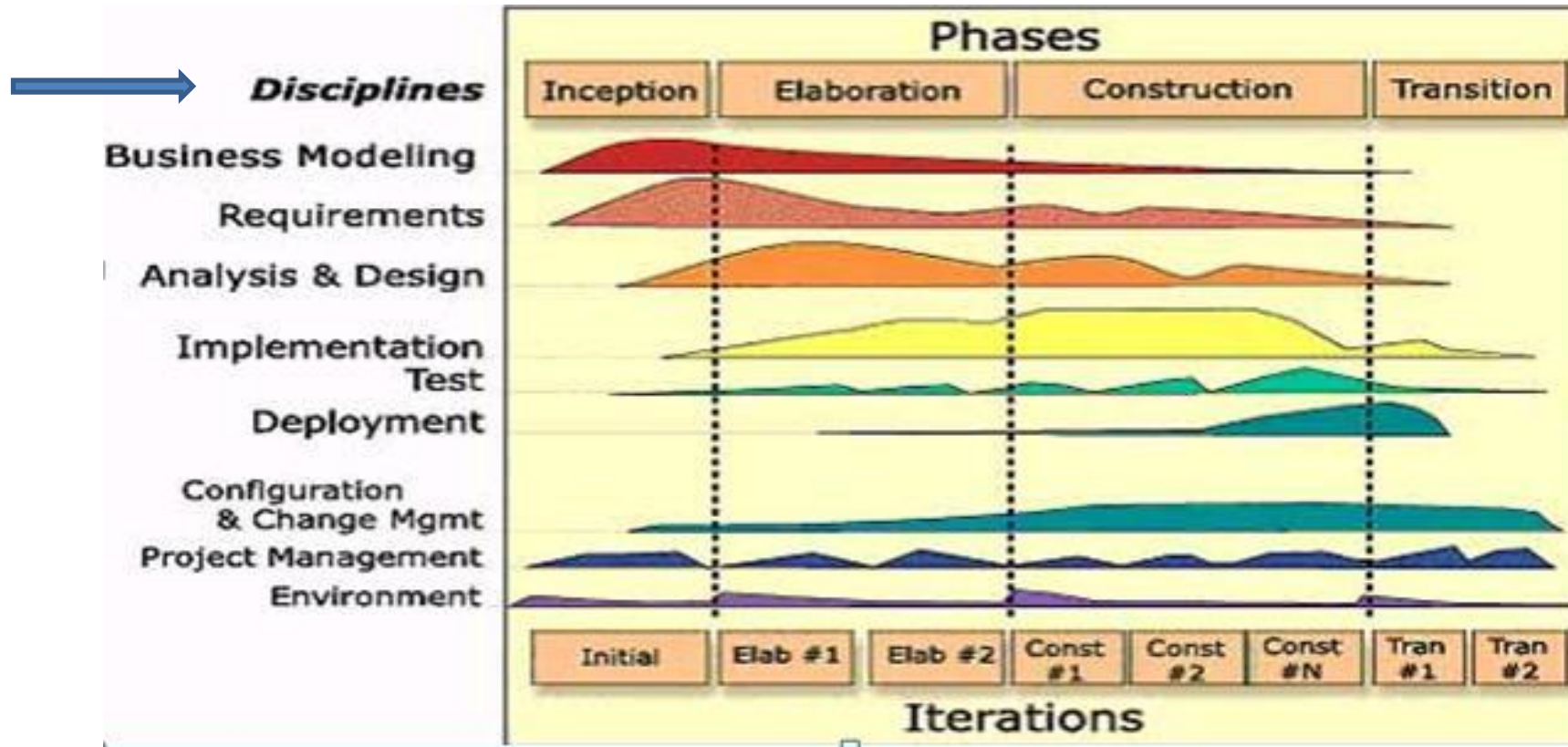


Обеспечивающие
системы (их
много!),
описанные минимально
как практики-
компоненты и проекты-
модули по изменению
целевой системы

Проблемы с ЖЦ 1.0

- Появилась параллельная инженерия (сначала «итерации», а потом и вообще всё смешалось)
- Нельзя обсуждать «почему *предприятие* вообще работает», методы работы, методологии. Только «как собрать из кусочков во времени».

Практики,
именованные по
дисциплинам!



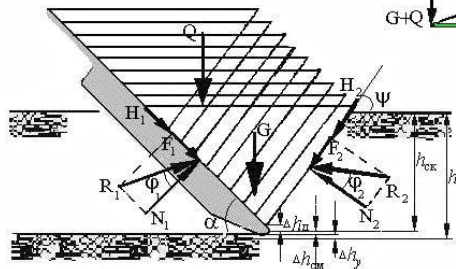
Горбатая диаграмма (hump diagram) из RUP (Rational Unified Process)

Какие компоненты предприятия? (как оно работает?)

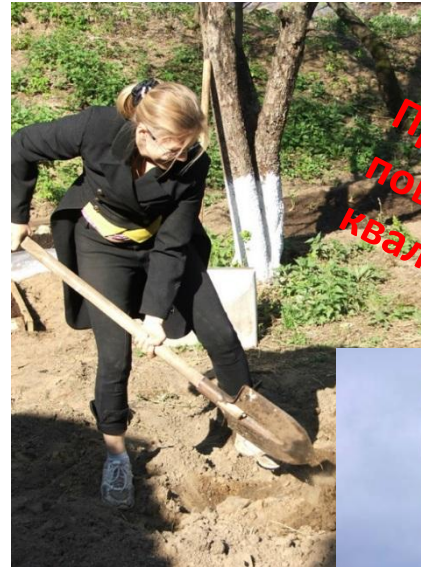
Практика = дисциплина + технология

Главное,
невидимое!

ВУЗ, ШКОЛА
(education)



Дисциплинированные (компетентные) исполнители [все ли люди?],



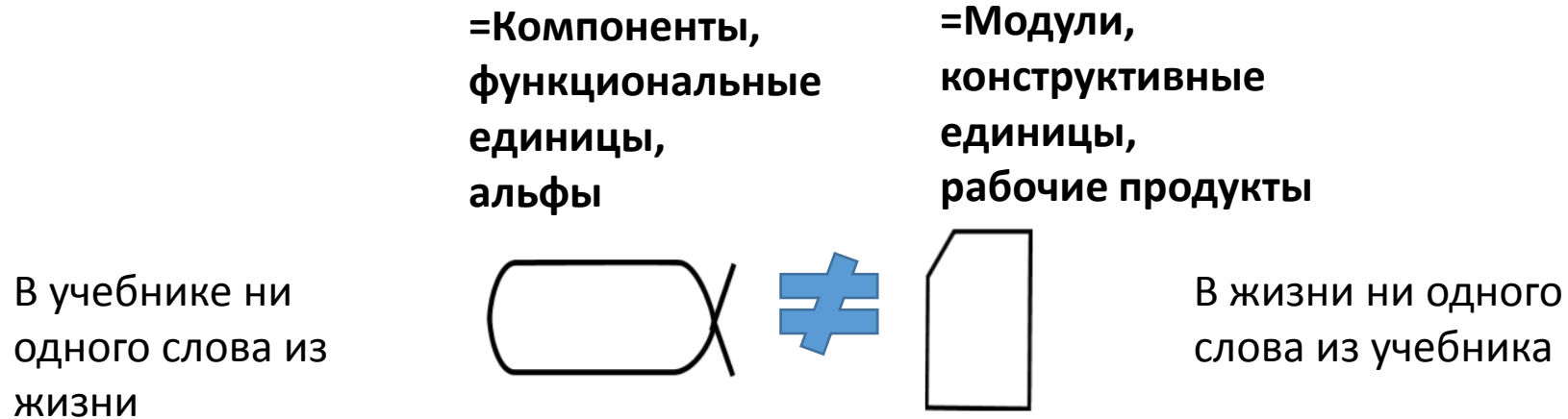
Производство,
повышение
квалификации (training)



Вторичное,
видимое!

обеспеченные необходимыми для поддержки дисциплины инструментами

Схематизация жизненного цикла: различение дисциплины и технологии в практике



- дисциплины = мышление (операции с абстрактными типизированными объектами). Меняются за 30 лет. Учатся в школе и ВУЗе. Практика определяется по её дисциплине. Инвестиции в «человеческий капитал».
- технологии = инструменты и рабочие продукты (поддержка мышления в экзокортексе). Меняются каждые 5 лет. Учатся на производстве. Проекты используют технологии как «ресурсы». Инвестиции в традиционный капитал.
- **Дисциплины тренируют на задачах (заранее поставленных)**
- **Понимание связи дисциплин и технологий, идеальных объектов дисциплин и рабочих продуктов в жизни нужно ТРЕНИРОВАТЬ, для этого обычно нужен преподаватель**
- **Маскировка в реальной жизни: а) задача не поставлена б) много отвлечений**

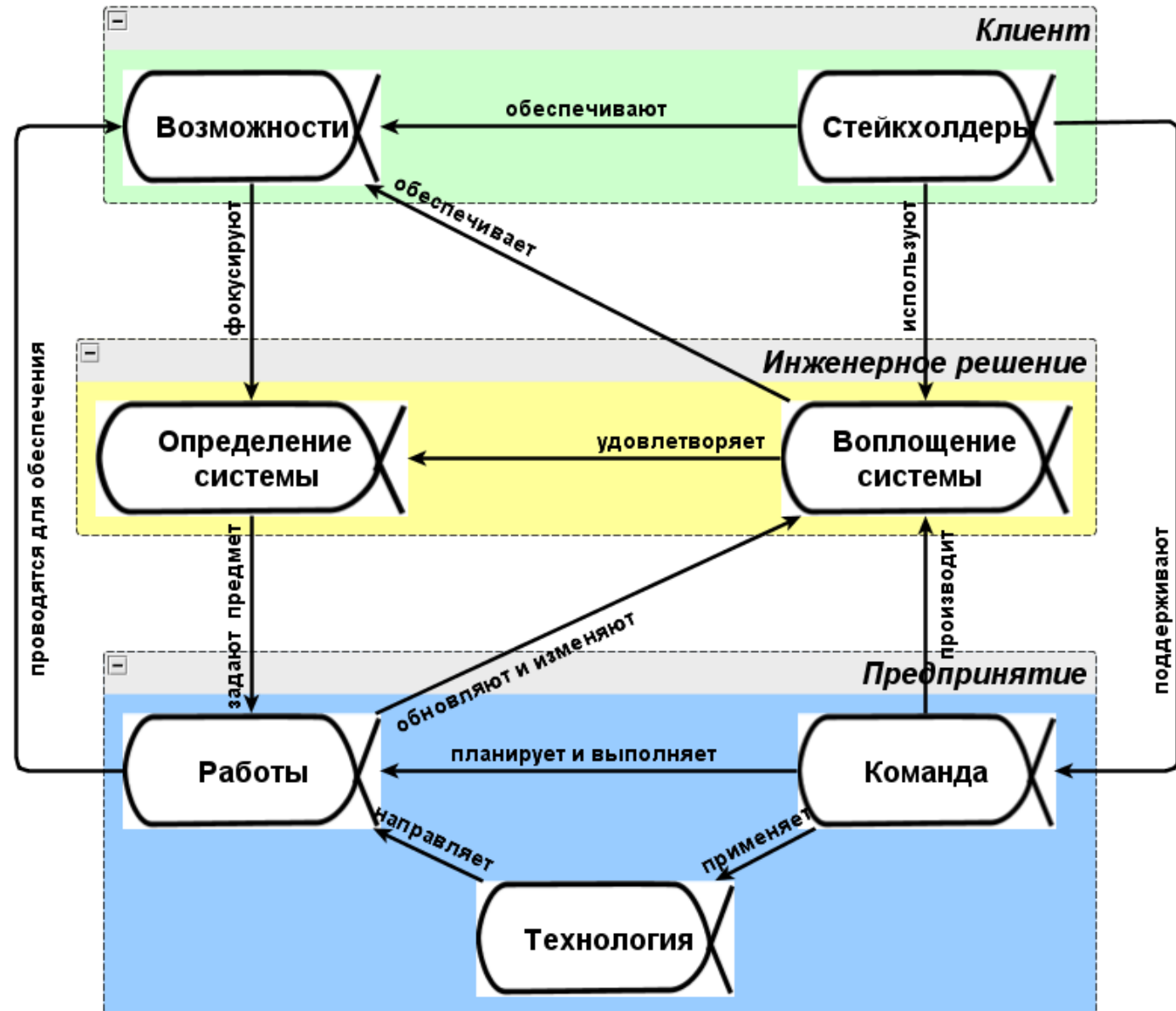
Системная схема проекта

Технологический менеджмент и предпринимательство
Using system

Инженерия
System of interest

Инженерный менеджмент
Enabling system

Технологический менеджмент



Ждём: machine learning specific processes

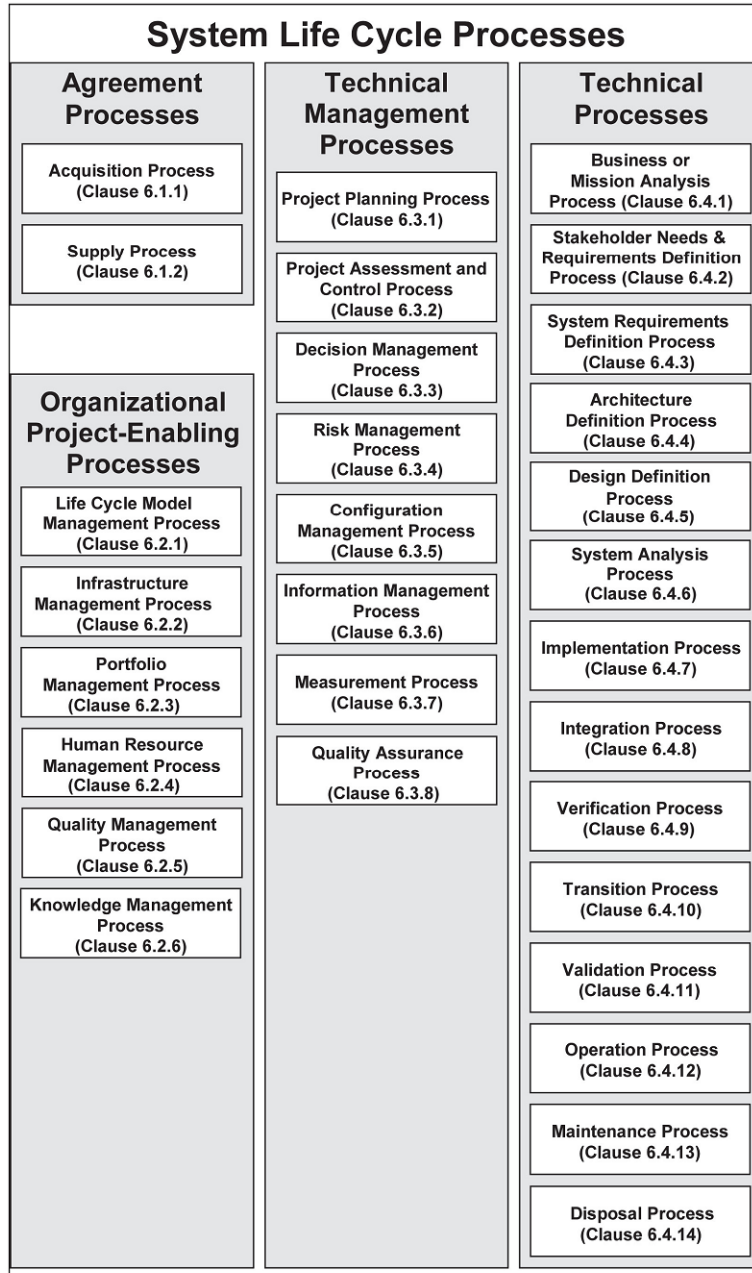


Figure 4 — System life cycle processes

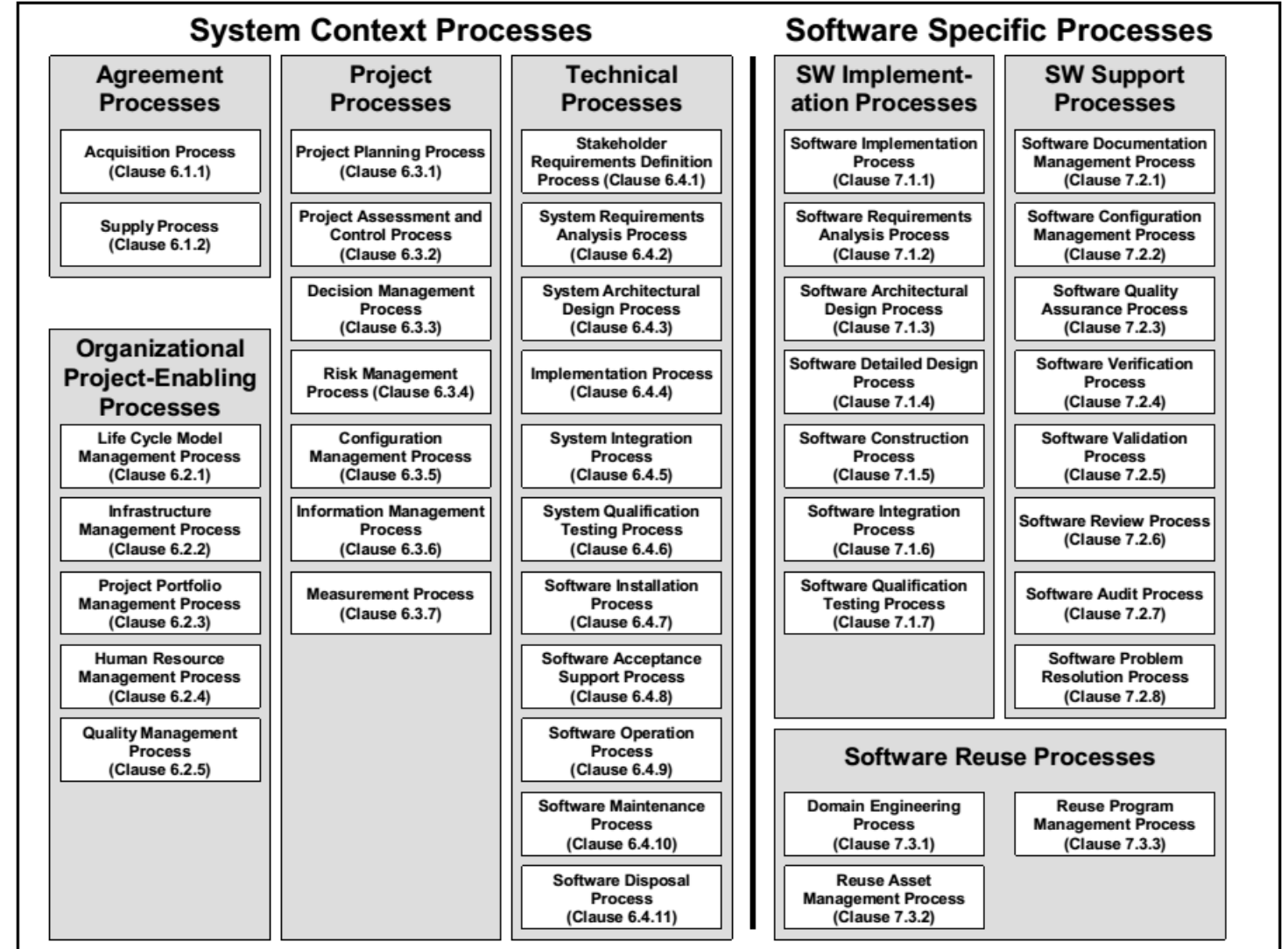


Figure 1 — Life Cycle Process groups

Какие практики можно ожидать?!

- Специализацию практик системной инженерии
- Специализацию практик программной инженерии

- Управление жизненным циклом (методология разработки)
- Инженерия требований
- Инженерия архитектуры
- Инженерия испытаний
- Управление конфигурацией
- ...

На многих уровнях технологического стека!

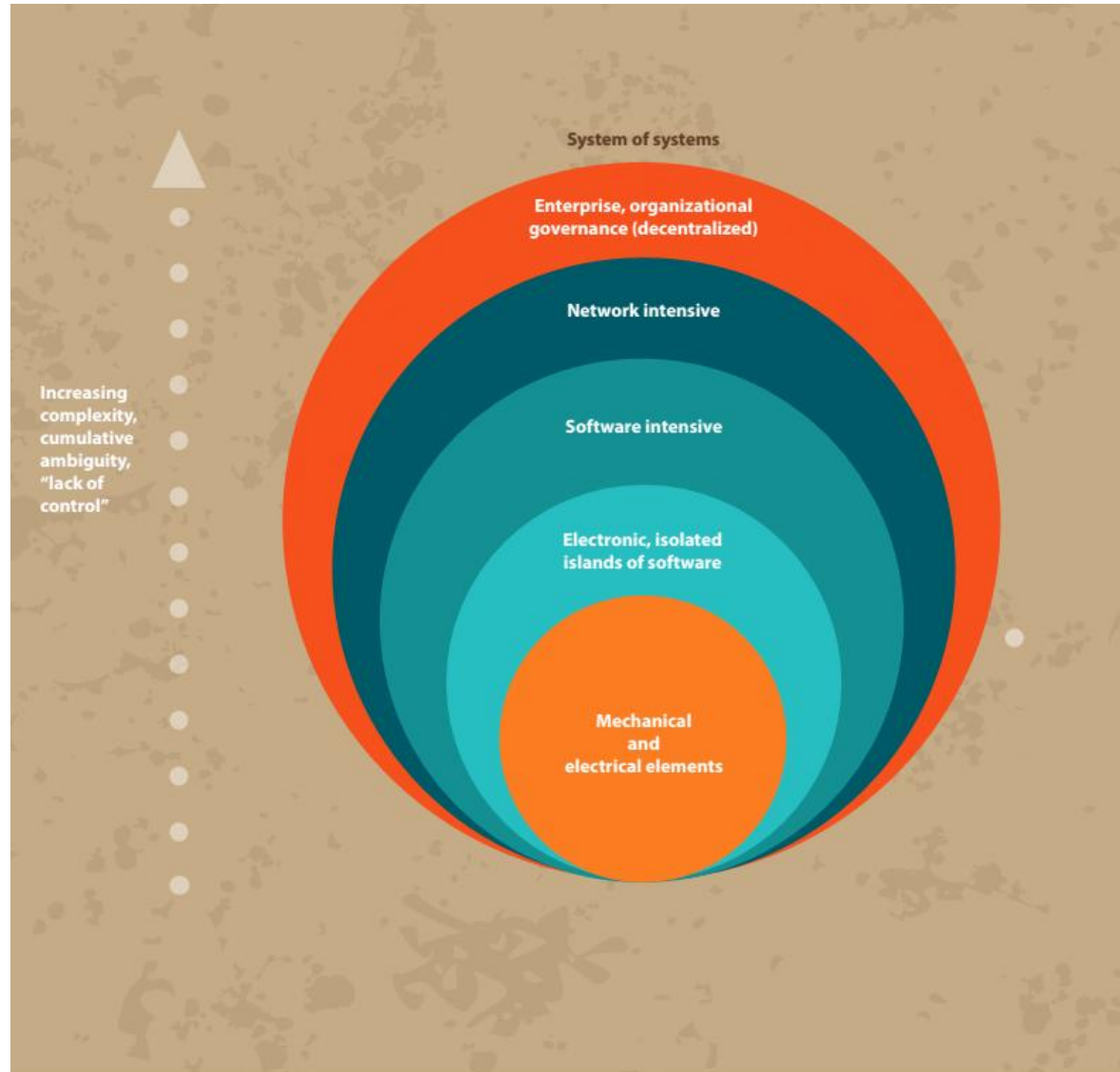
Жизненные циклы обучающихся систем

Инженерия психики	Инженерия машинного обучения	Инженерия предприятия	Системная инженерия
Постановка задачи на модернизацию (всегда brownfield)	Замысел и требования	Стратегирование	Замысел и требования
Намеревание	Архитектура	Архитектура	Архитектура и проектирование
(выращивание субстрата)	программирование архитектуры	Постановка практик: набор персонала, закупка технологий	Изготовление и интеграция/сборка
Дообучение (ибо всегда brownfield)	Обучение (training),	Постановка практик в части обучения работе	наладка
Автоматизация навыка	Передача обучения (transfer learning), сжатие сетки, ансамблирование (но ярко выраженной сборки нет, плохая модульность)	Merge/aquisition на уровне предприятий (но нет «сборки» отдельных практик, плохая модульность)	Модернизация заменой модулей
Аттестация, экзамены, освидетельствование	Проверка и приёмка	Оценка, аттестация	Проверка и приёмка
Жизнь	Вывод (inference)	Работа	Эксплуатация

Жизненный цикл систем машинного обучения

- **Экстремальный ЖЦ:** начинается много раньше, чем замысел, заканчивается много позже вывода из эксплуатации
- **Многоуровневый ЖЦ:** разный на всех уровнях интеллект-стека
- **Нет методологий разработки:** сказать agile – это ничего не сказать (кто признается, что он «негибкий?»), нет отдельных практик, принципов, инструментов
- **Нет учебников по практикам ЖЦ, только отдельные статьи.**

INCOSE VISION 2025



Проблемы с модульностью на высоких уровнях (где появляются сети, люди – непонятная связность): переход от сборки-наладки к «эволюции», «обучению», «системообразованию»

Метафоры жизненного цикла машинного обучения



часовщик



садовник



Обучение Tesla – уже сегодня

лесник

инженерия

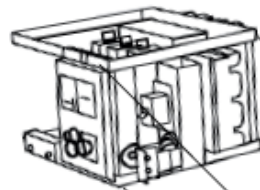
Восток: смириться и растить. Запад: добиваться модульности, растить потом.

МЕЖДИСЦИПЛИНАРНОСТЬ

Интеллект-функции + интеллект-стек + где это в мире



-Модули

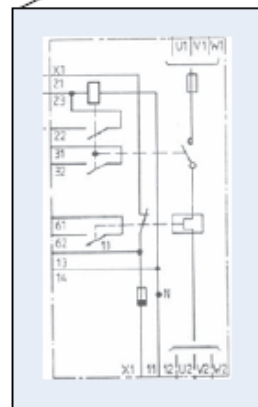


Ex:
-M87-K5

The product aspect is used to highlight the constructional relations (assembly) of the components of the object.

The function aspect is used to highlight the functional relations among the components of the object.

Ex:
=S12=16



=Компоненты

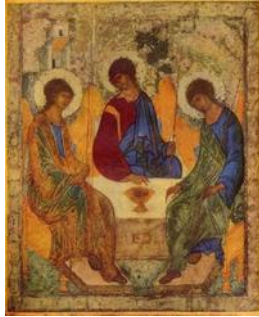
+Места

Ex:
+D2+Z1

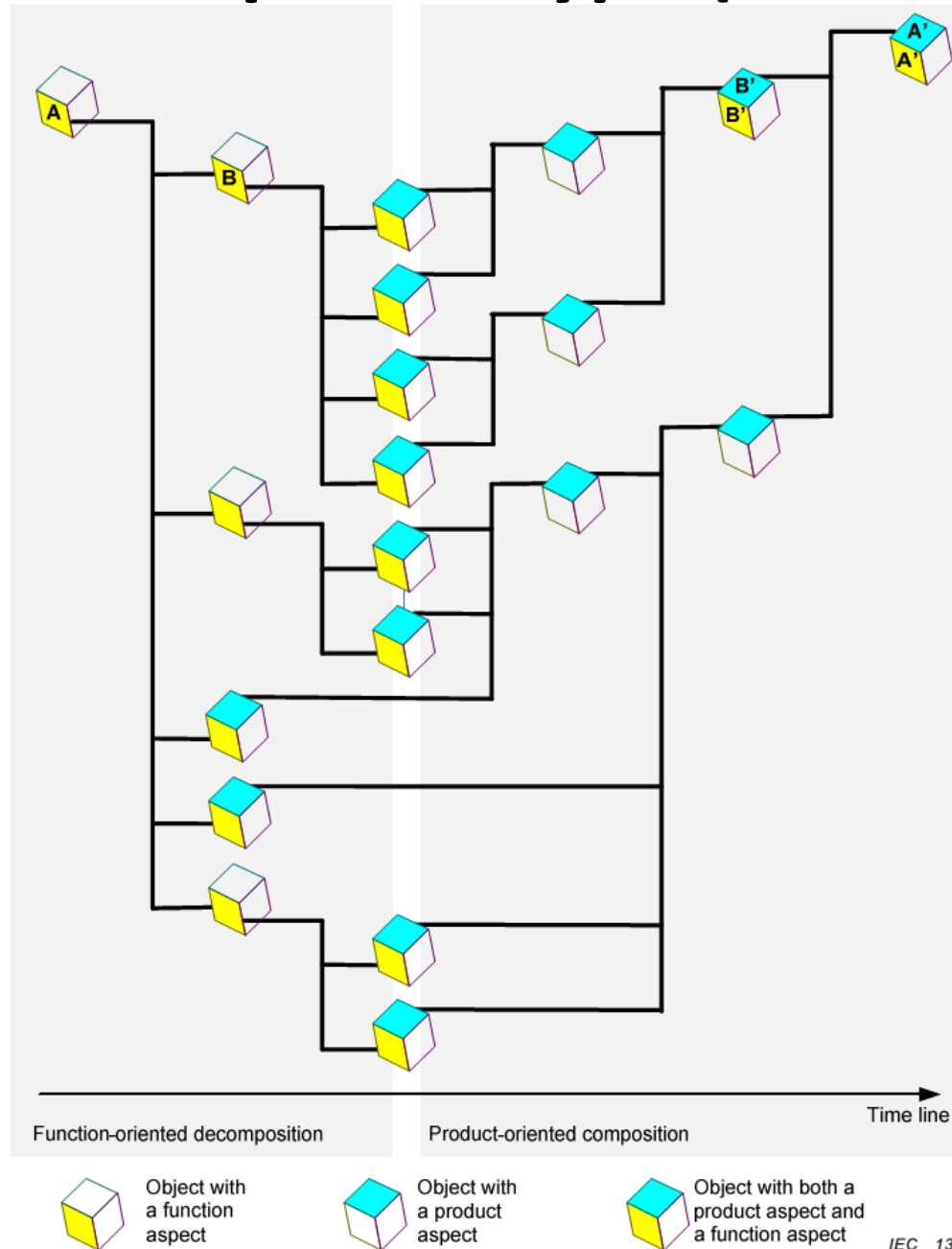
The location aspect is used to highlight the spatial relations among the components of the object.



Главная дисциплина: архитектура (важные решения)



«Логическая архитектура»
(функциональная
декомпозиция, структура
компонент) итеративно
совмещается с «физической
архитектурой» (продуктная
композиция, структура
модулей)



Модульный синтез:

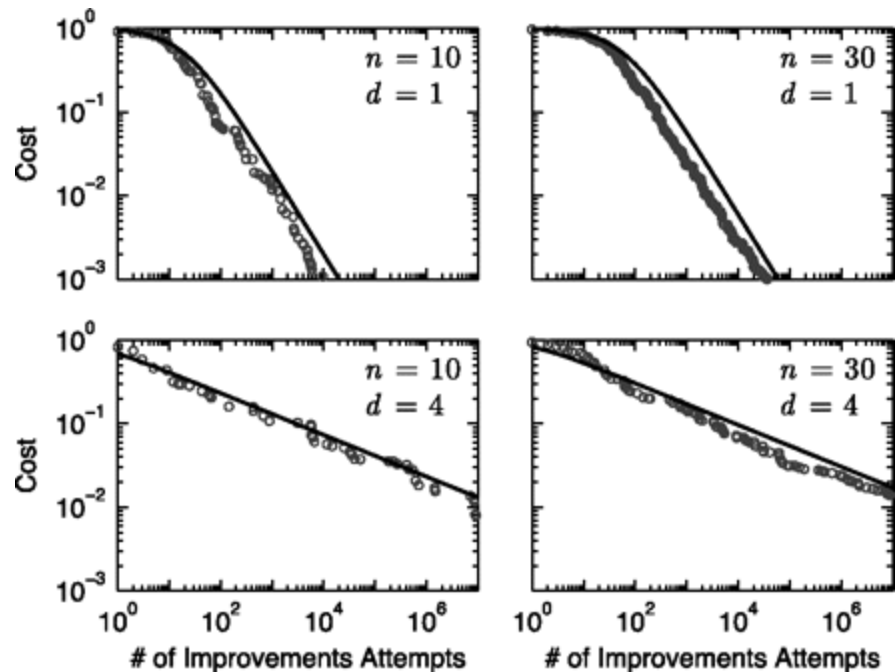
- Компонуемость – собираемость модулей
- Комозициональность – собираемость функций на модулях

<https://pages.nist.gov/cpspwg/>

ISO 81346-1
Figure 7

Малая связность: ключ к развитию и совершенствованию

- Модульность: каждая связь имеет цену. Не было бы цены, не было бы модулей
 - <http://arxiv.org/abs/1207.2743>
- Меньше связность – круче улучшения!
 - <http://www.pnas.org/content/108/22/9008.full>



Падение стоимости при улучшении отдельных модулей (n), при разном числе связей каждого из них (d)

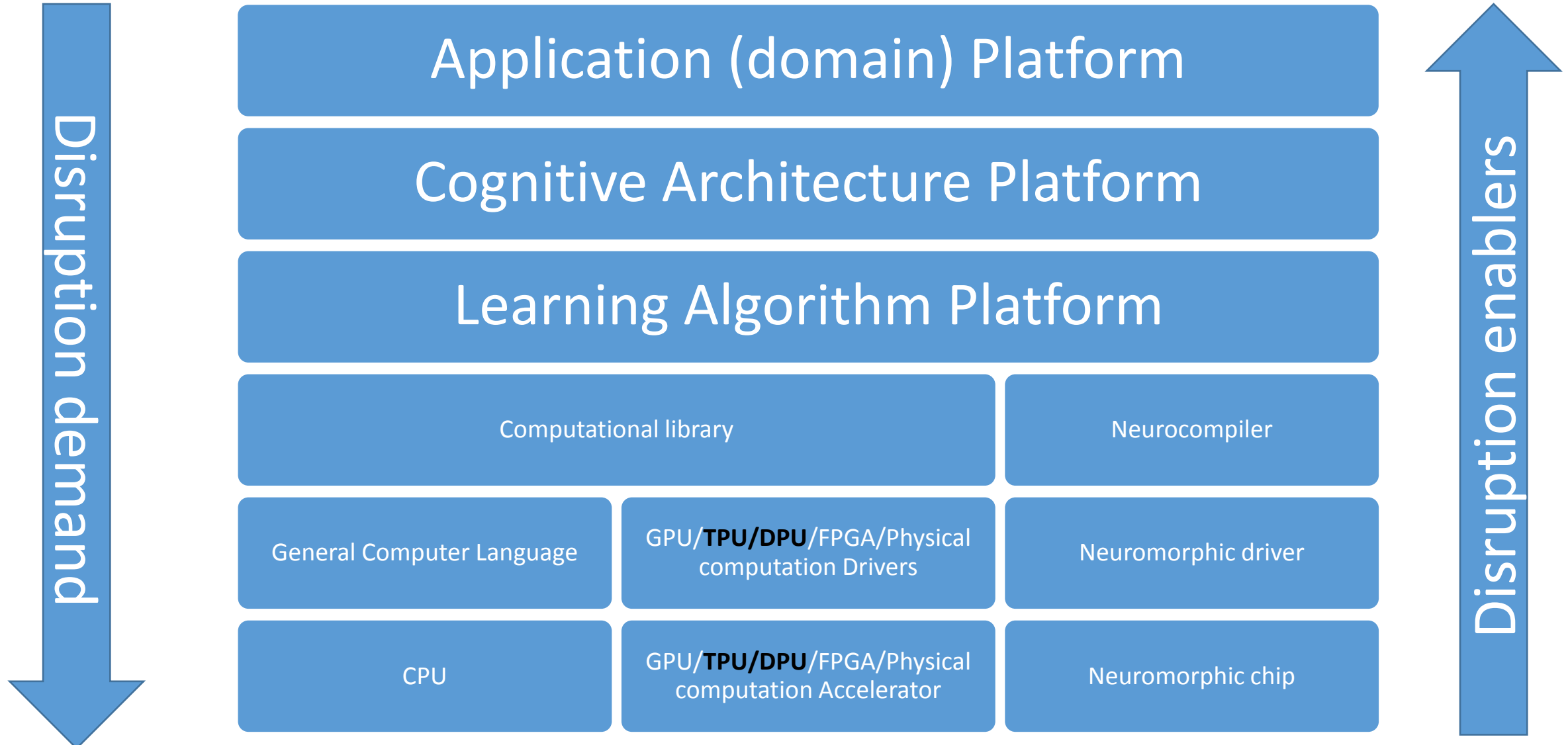
Чтобы изменения улучшали дело, нужно меньше связей!

Модульности очень разные:

- Knowledge graphs (онтологии и семантики) – структурированная память (MemNN, differentiable neural computer)
- Embeddings – priors (замечание Nando de Freitas, <http://ailev.livejournal.com/1240509.html>)
- Models as modules (например, progressive networks для multitask learning, ансамблирование и его моделирование -- <https://arxiv.org/abs/1605.06431>, разные виды transfer learning)
- Learning algorithms as modules (например, frameworks), и отдельные элементы для алгоритмов (обзор -- <http://fastml.com/deep-learning-architecture-diagrams/>).
- ...

Серебряной пули нет, и никогда не будет.

Intelligence Platform Stack

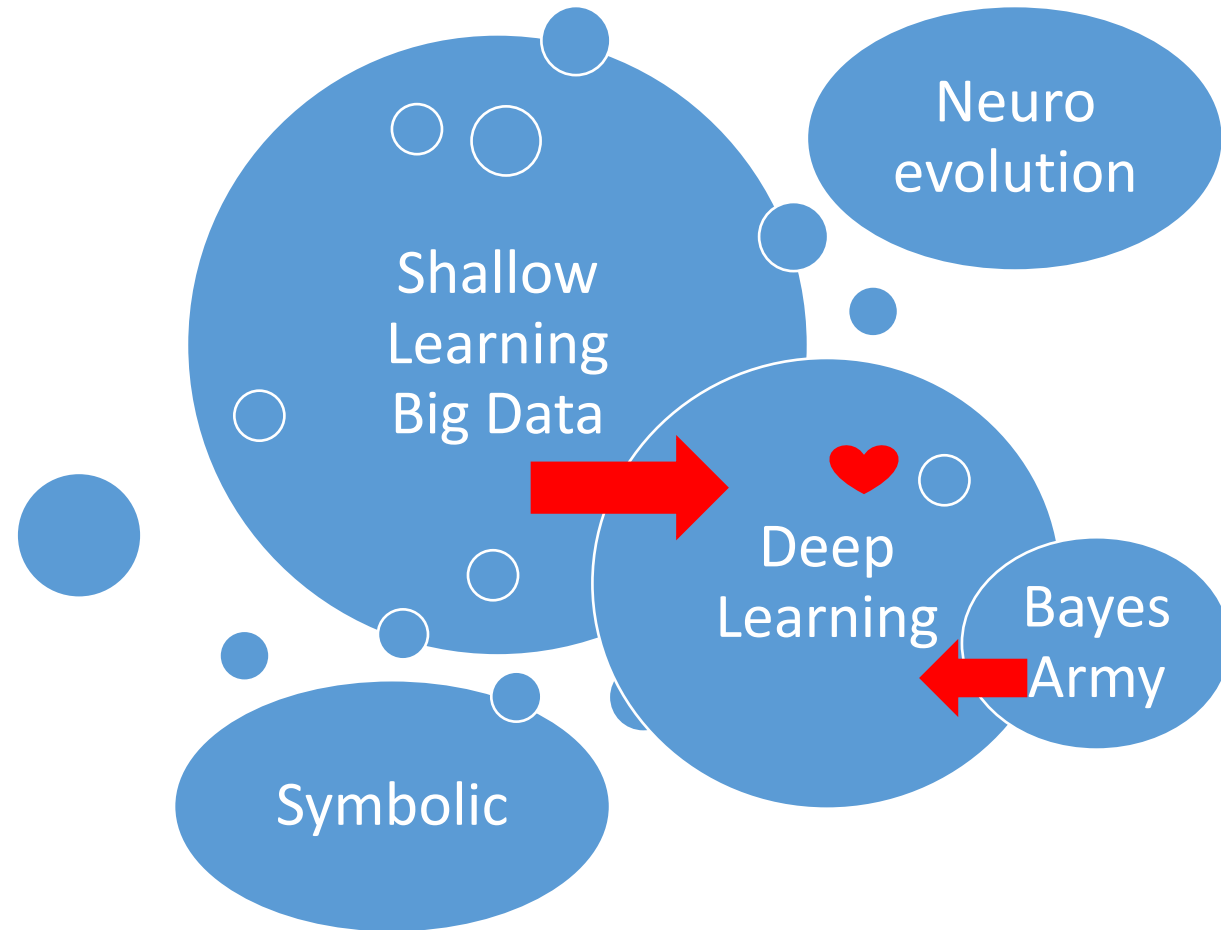


**Thanks for computer gamers for their disruption demand
to give us disruption enabler such as GPU!**

Платформы машинного обучения

- Одной платформы не хватит никогда!
- Master algorithm: тренд на **гибридизацию** разных подходов
- Когнитивная архитектура: тренд на **комбинирование** разных подходов
- Reinforcement learning
- Adversarial architectures

Теорема бесплатного завтрака



Conversion of *engineerings* and Disruption of *engineerings*

Janosh Szepanovits. **Convergence:** Model-Based Software, Systems And Control Engineering

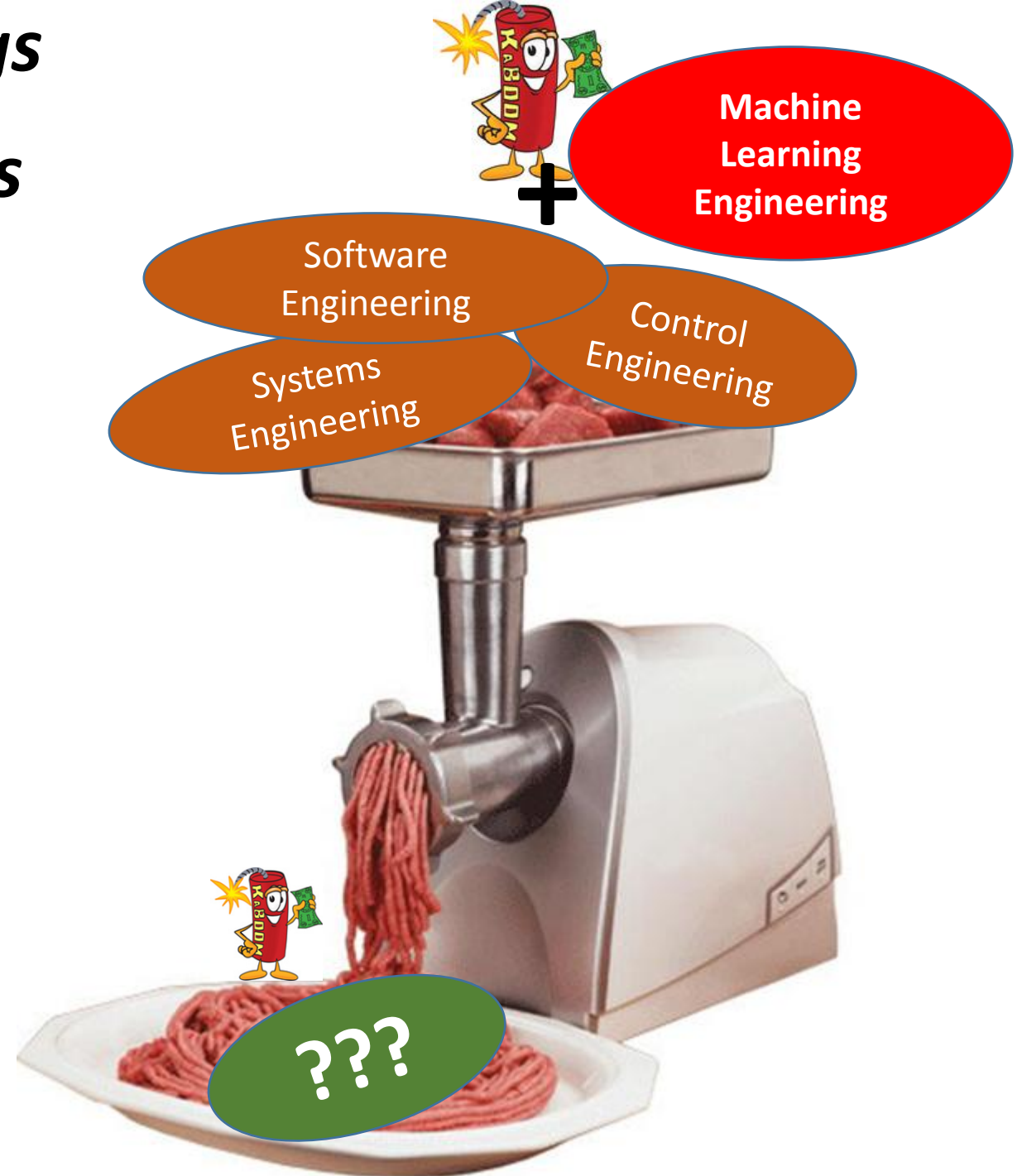
<http://www.infoq.com/presentations/Model-Based-Design-Janos-Sztipanovits>

Le Bottou – «Machine Learning **disrupts** software engineering»

<http://leon.bottou.org/slides/2challenges/2challenges.pdf>

We can add:

- Machine learning disrupts systems engineering
- Machine learning disrupts control engineering
- ...
- **Machine learning disrupts contemporary engineering**



Литература по инженерии MLE: больше постановки проблем, чем решения

- Leon Bottou, **Two big challenges in machine learning**
<http://leon.bottou.org/slides/2challenges/2challenges.pdf>
- D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, Dan Dennison, **Hidden Technical Debt in Machine Learning Systems**
<http://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf>
- Nando de Freitas, **Learning to learn and compositionality with deep recurrent neural networks**
<https://youtu.be/x1kf4Zojtb0>
- Diogo Almeida, **Modular in theory, inflexible in practice**
<http://conferences.oreilly.com/artificial-intelligence/ai-deep-learning-bots-ny/public/schedule/detail/54081>

Can we use systems and software engineering wisdom in MLE?

**Engineering is not only about modularity and modular synthesis!
What about other aspects?!**

- More attention to left part of V-diagram (as in MBSE)
- More attention to right part of V-diagram (DevOps, release early, release often – optimizations later)
- ...
- **What else?**

Спасибо за внимание!

Анатолий Левенчук,

<http://ailev.ru>

ailev@asmp.msk.su

TechInvestLab