

A=F(?): Как не следовать здравому смыслу и гробить проекты

Евгений Кривошеев,
ekrivosheev@scrumtrek.ru

Проектирование систем как принятие решений

- Квадратики и стрелочки?
- Дизайн – это компромисс
- Контекст важен – это ограничения для дизайна
- Нужно принять много решений, например↓

Когда остановиться
рисовать квадратики
и начать $x e^{**}$ читать?

Что такое архитектура?

- Внезапно, рекурсивное определение через требования
- Внезапно, определение через риски
- От чего зависят эти наши решения? $A=F(?)$

$A = F(R, q)$: Формальный подход

$A = F(\text{outer functional requirements, outer quality attributes})$;



Agility? Never heard.



$A = F(ORq, IRq)$: Менее формально

```
A = F(  
  outer functional requirements,  
  outer quality attributes,  
  
  inner functional requirements, //Призрачная угроза  
  inner quality attributes, //Разрабы тоже люди^W юзеры  
  design guidelines & code style //Регламенты пишутся кровью  
);
```

$A = F(Rq, ?)$: Что не
учитываем и гробим
проекты

A=F(Rq,?): Что не учитываем

```
A = F(  
  outer FR & QA,  
  inner FR & QA, //Разрабы тоже люди^W юзеры  
  design guidelines & code style, //Регламенты пишутся кровью  
  
  personal pain experience //Какие архипотезы проверять ASAP  
);
```

A=F(Rq,?): Что не учитываем

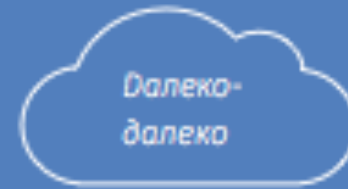
```
A = F(  
  outer FR & QA,  
  inner FR & QA, //Разрабы тоже люди^W юзеры  
  design guidelines & code style, //Регламенты пишутся кровью  
  personal pain experience //Какие архипотезы проверять ASAP  
  
  trust to team //Сделаем на монадах!  
);
```

A=F(Rq, ?): Что не учитываем

```
A = F(  
  outer FR & QA,  
  inner FR & QA, //Разрабы тоже люди^W юзеры  
  design guidelines & code style, //Регламенты пишутся кровью  
  personal pain experience //Какие архгипотезы проверять ASAP  
  trust to team, //Сделаем на монадах!  
  
  team & company structure //Закон Конвея  
);
```

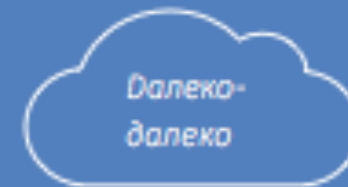
Менеджмент

решение экономической задачи
«сроки-ресурсы-скоуп-риски»



Продуктовый менеджмент

«что будет завтра?»



Команда разработки 1
Component team vs Feature team

Команда разработки 1
Component team vs Feature team



A=F(Rq,?): Что не учитываем

```
A = F(  
  outer FR & QA,  
  inner FR & QA, //Разрабы тоже люди^W юзеры  
  design guidelines & code style, //Регламенты пишутся кровью  
  personal pain experience //Какие архгипотезы проверять ASAP  
  trust to team, //Сделаем на монадах!  
  team & company structure, //Закон Конвея  
  
  resources: {time, people, money, partners, trust↑, risks}  
);
```

A=F(Rq,?): Что не учитываем

```
A = F(  
  outer FR & QA,  
  inner FR & QA, //Разрабы тоже люди^W юзер  
  design guidelines & code style, //Регламенты пишутся кровью  
  personal pain experience //Какие архгипотезы проверять ASAP  
  trust to team, //Сделаем на монадах!  
  team & company structure, //Закон Конвея  
  resources: {time, people, money, partners, trust↑, risks},  
  
  внутрикорпоративные ограничения  
);
```


A=F(Rq,?): Что не учитываем

```
A = F(  
  outer FR & QA,  
  inner FR & QA, //Разрабы тоже люди^W юзер  
  design guidelines & code style, //Регламенты пишутся кровью  
  personal pain experience //Какие архгипотезы проверять ASAP  
  trust to team, //Сделаем на монадах!  
  team & company structure, //Закон Конвея  
  resources: {time, people, money, partners, trust↑, risks},  
  внутрикорпоративные ограничения,  
  
  integration landscape, law issues, locked-in vendors, ...  
);
```

Вопрос #1: Какие из факторов Вы честно учитываете при проектировании?

Вопрос #2: **OMG!** Их
же невозможно учесть
все!

Смотрим по важности

- Свободно доверяйте и делегируйте
- Свободно откладываете
- Превед Lean Development :)

Применяй меня полностью

- Поймите бизнес-ограничения
- Формальные регламенты vs написанные кровью
- Не доверяйте себе в важных решениях
- Пользуйтесь шпаргалкой факторов
- Приоритезируйте факторы
- Откладывайте решения

