



Software Engineering Conference Russia
2018

Много методологий в одном
проекте.
Как и зачем?

Банальности, проверенные опытом

Дельгадо Филипп, NEXIGN

Методология



Методология

Процессы

Роли

Артефакты

Ценности

Практики

...

Алистер Коуберн

«Каждому проекту –
свою методологию»

<http://alistair.cockburn.us/>



Каждому проекту – своя методология

Чем больше размер команды, тем «больше» нужна методология

Каждому проекту – своя методология

Чем больше размер команды, тем «больше» нужна методология

Чем опаснее ошибки, тем «плотнее» нужна методология

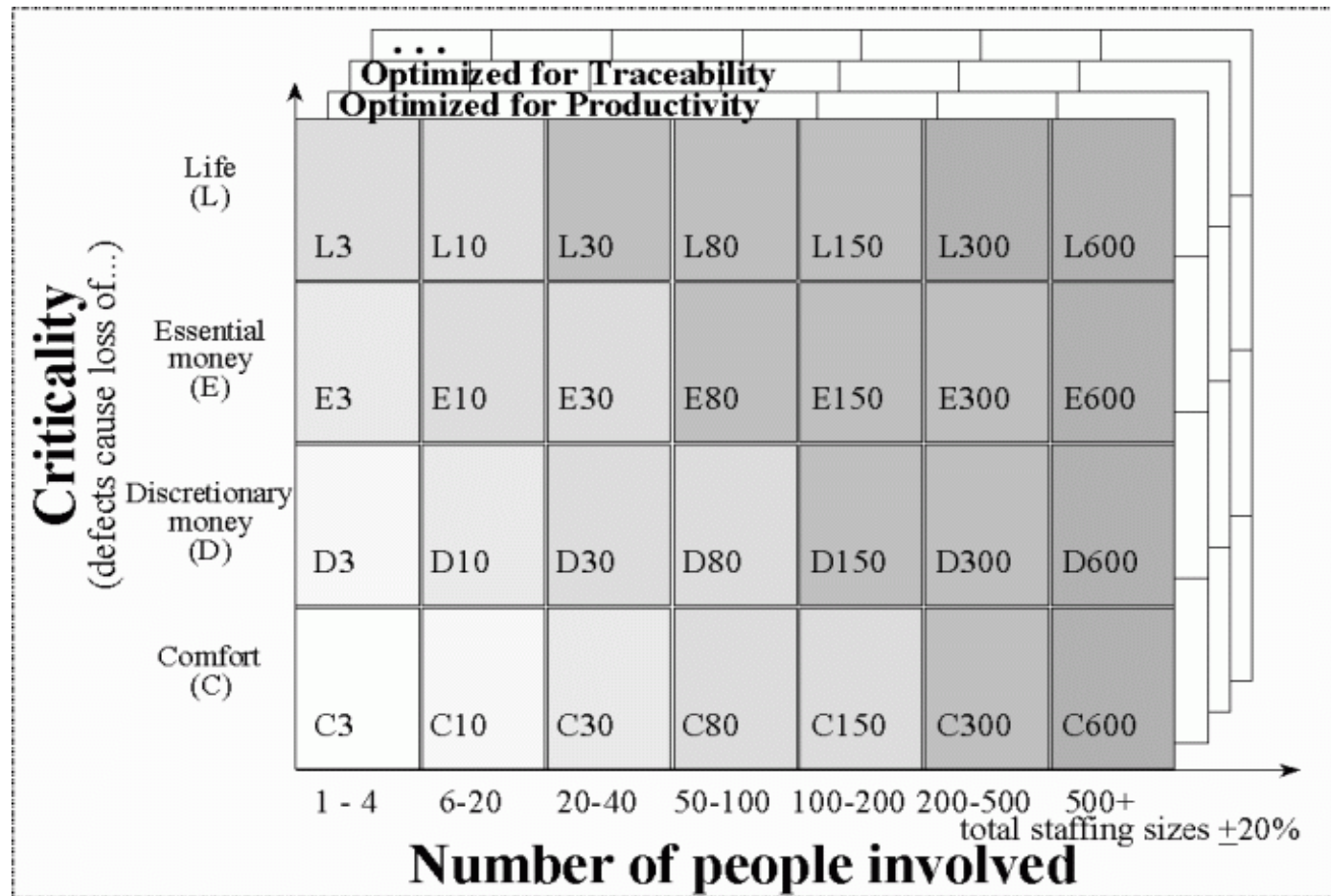
Каждому проекту – своя методология

Чем больше размер команды, тем «больше» нужна методология

Чем опаснее ошибки, тем «плотнее» нужна методология

Незначительное увеличение «размера» или «плотности» стоит дорого

Каждому проекту – своя методология



Каждому проекту – своя методология

Разная критичность

Разный размер команды

Каждому проекту – своя методология

Разная критичность

Разный размер команды

Разные люди

Каждому проекту – своя методология

Разная критичность

Разный размер команды

Разные люди

Разные цели

Каждому проекту – своя методология

Разная критичность

Разный размер команды

Разные люди

Разные цели

предсказуемость

time-to-market

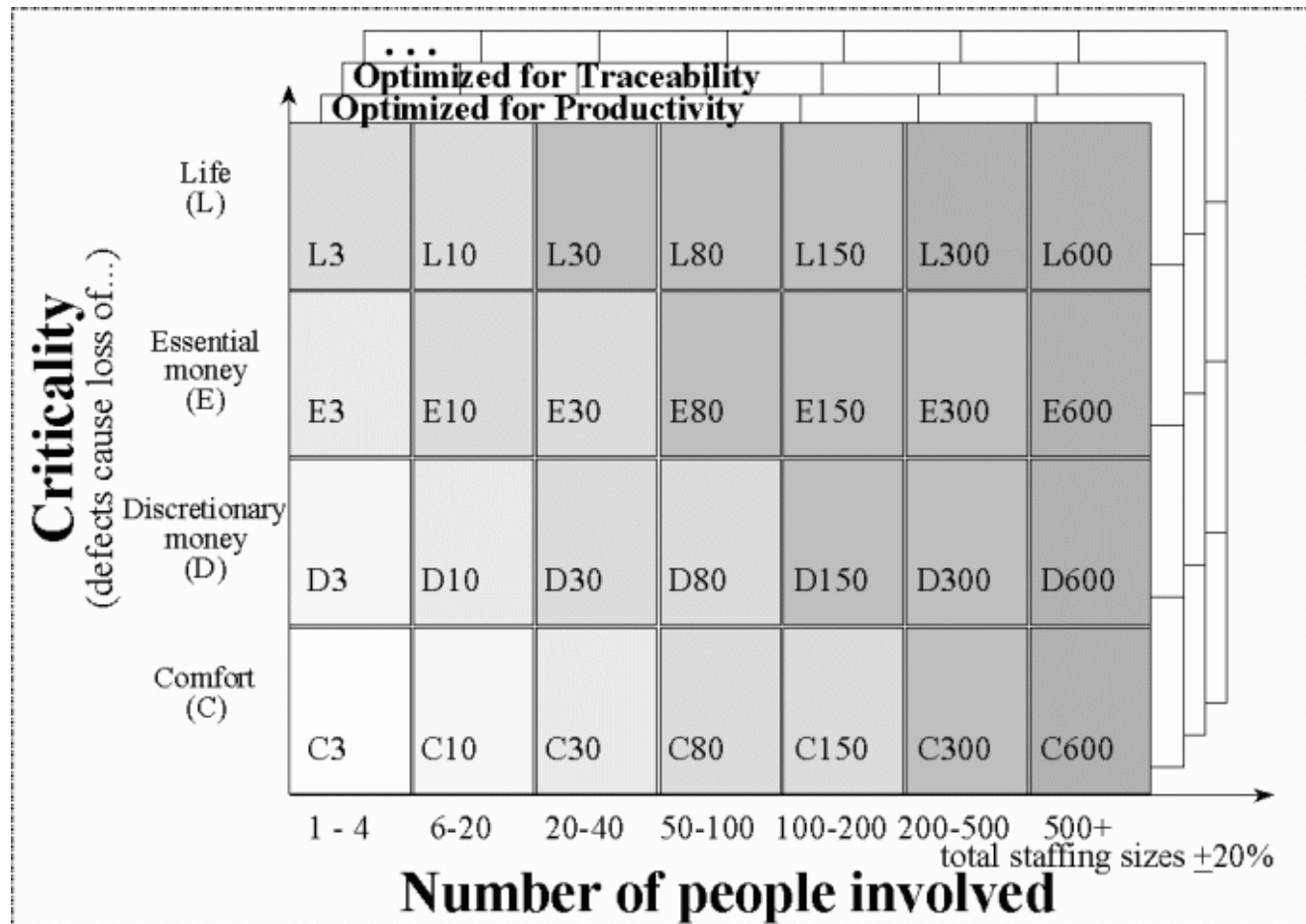
контролируемость

модность

Например

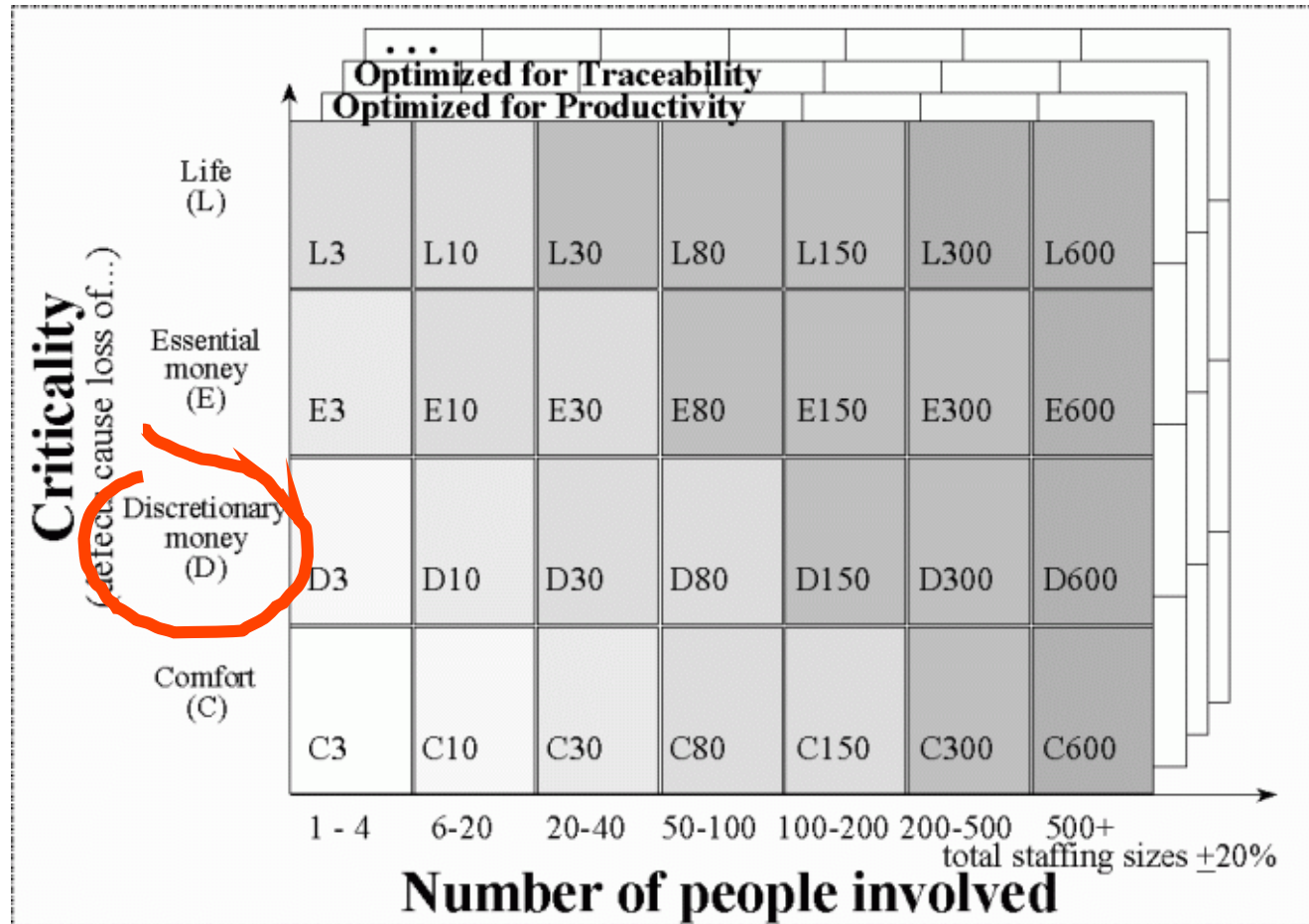
Подберем методологию для платежной системы

Например



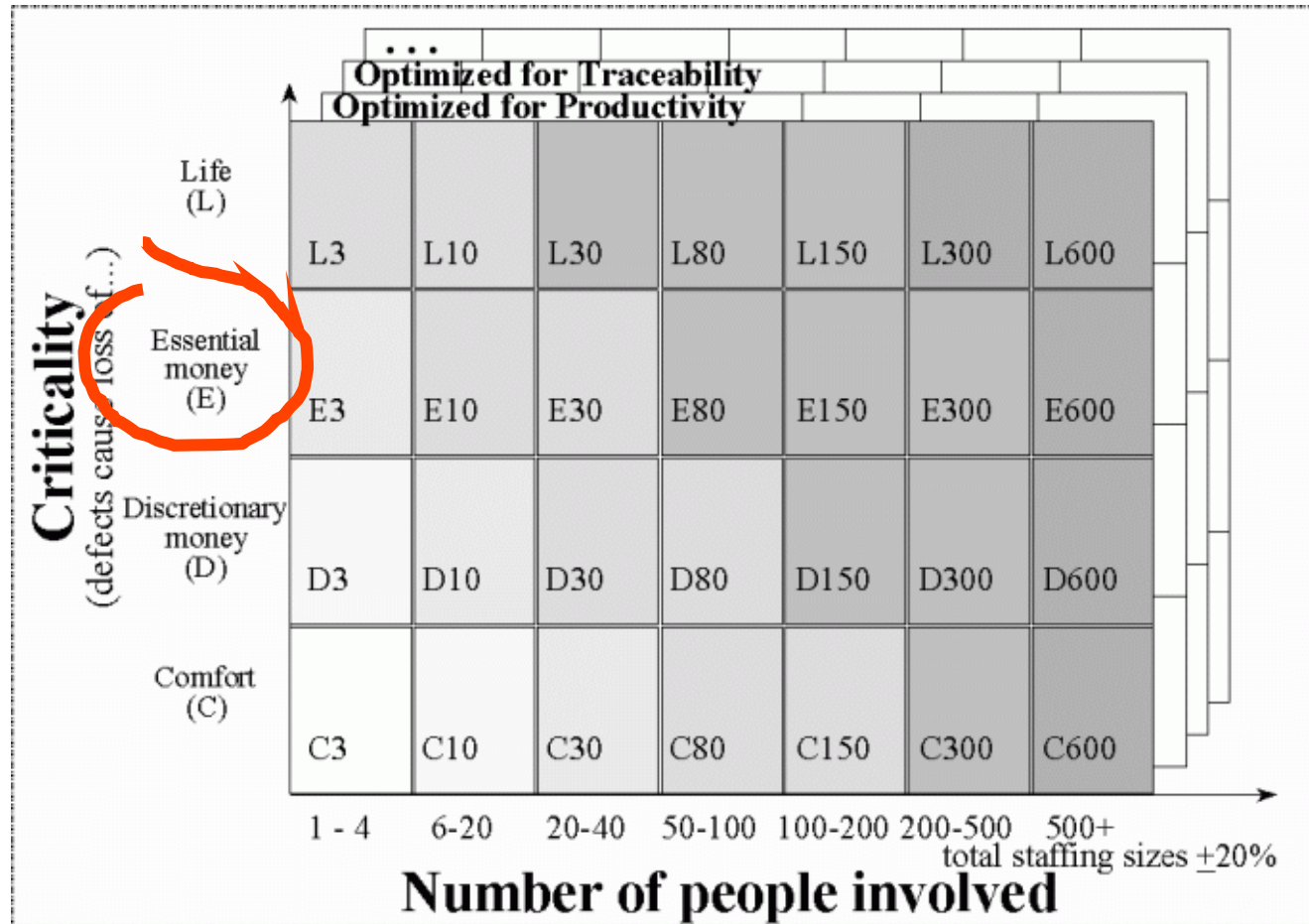
Например

Платежная система

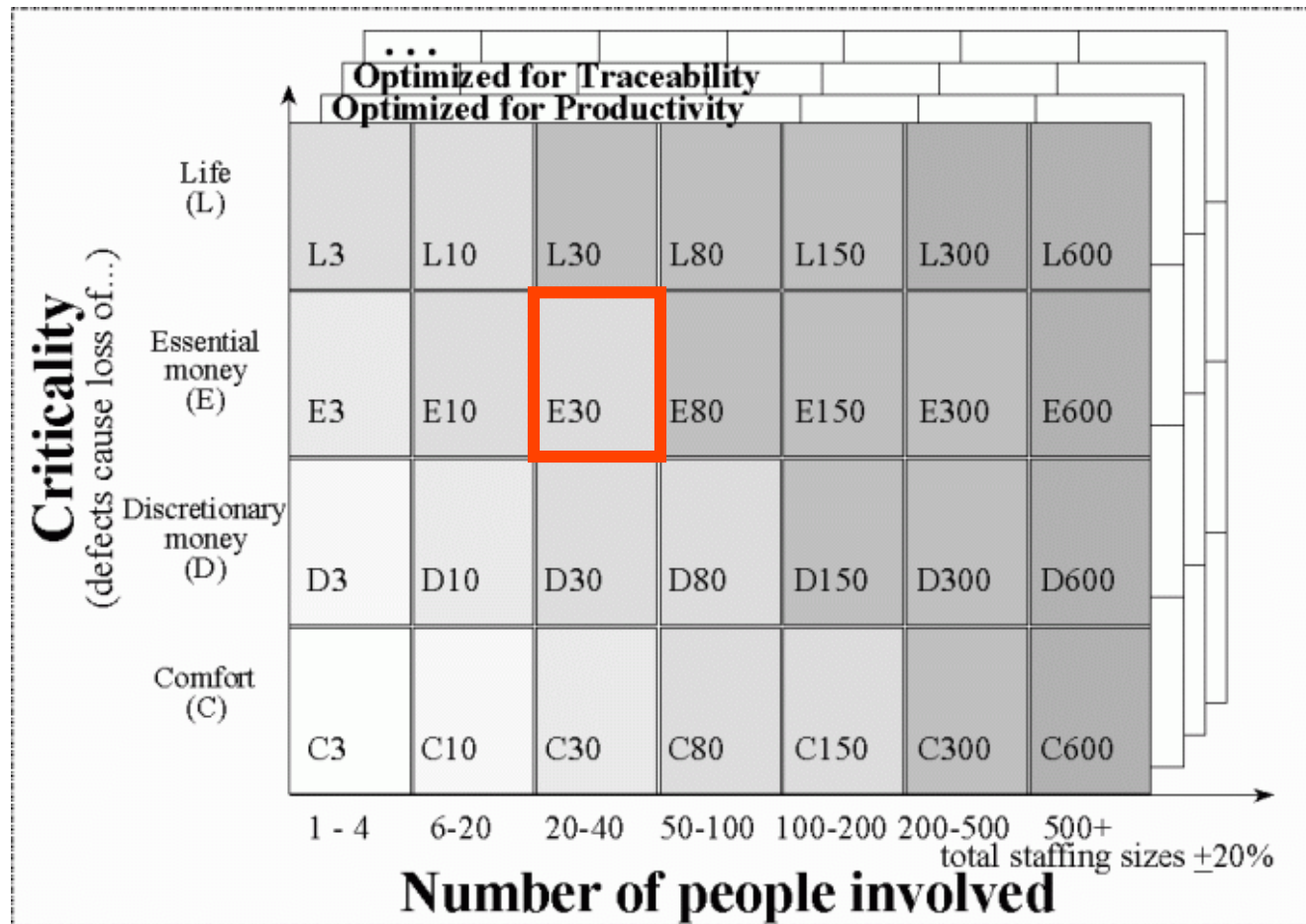


Например

Платежная система
Для банка



Например

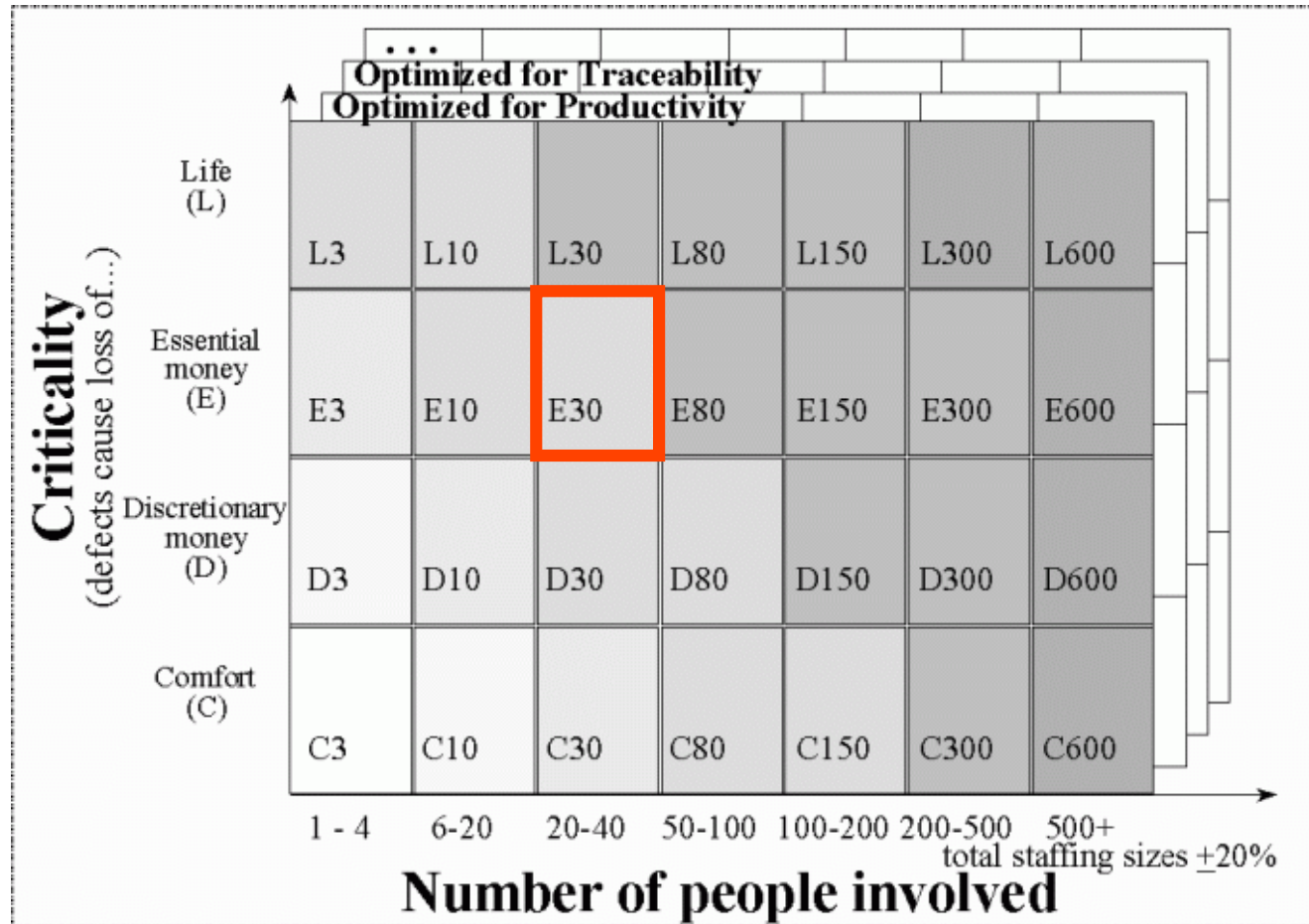


Платежная система

Для банка

Команда 20+ человек

Например



Платежная система

Для банка

Команда 20+ человек



Едим слона по частям



Например

Не весь проект критичен

Проверка на отмывание денег

Работа к банковскими картами

Хранение персональных данных

Разделим на части

Не весь проект критичен

Проверка на отмывание денег

Работа к банковскими картами

Хранение персональных данных

Выделяем критичные модули

Разделим на части

Не весь проект критичен

- Проверка на отмывание денег

- Работа к банковскими картами

- Хранение персональных данных

Выделяем критичные модули

Формируем отдельную методологию работы

- Только senior developers

- PCI DSS Audit

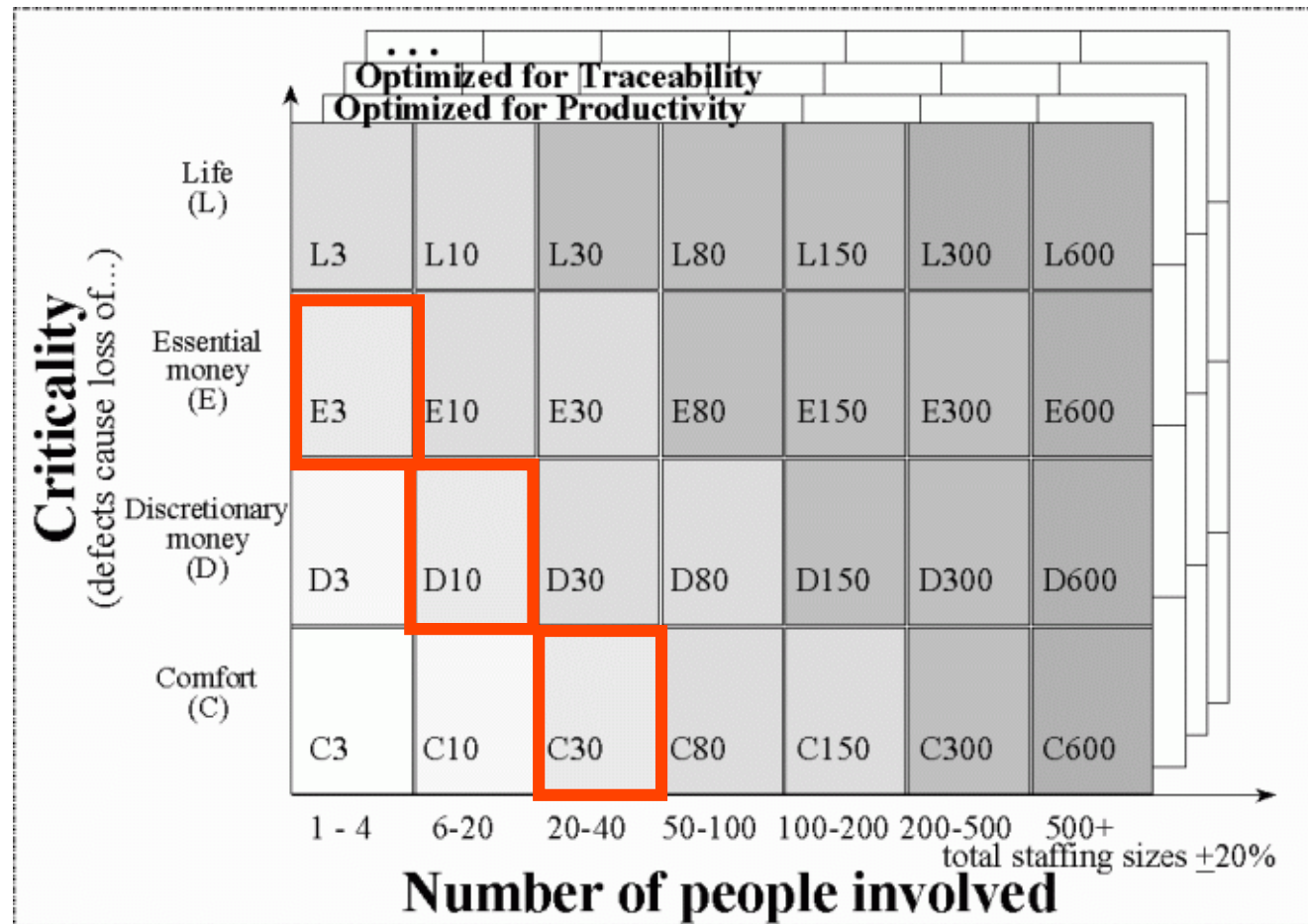
- Документирование

- Двойной Code Review

- Специальный процесс выкладки

Получим

Элементы разных методологий



В развитии

А что дальше?

Начало



Начало

Нет проверенных постановок

Исследовательские задачи

Простота изменений

Маленькая команда

Начало

Нет проверенных постановок

Быстро

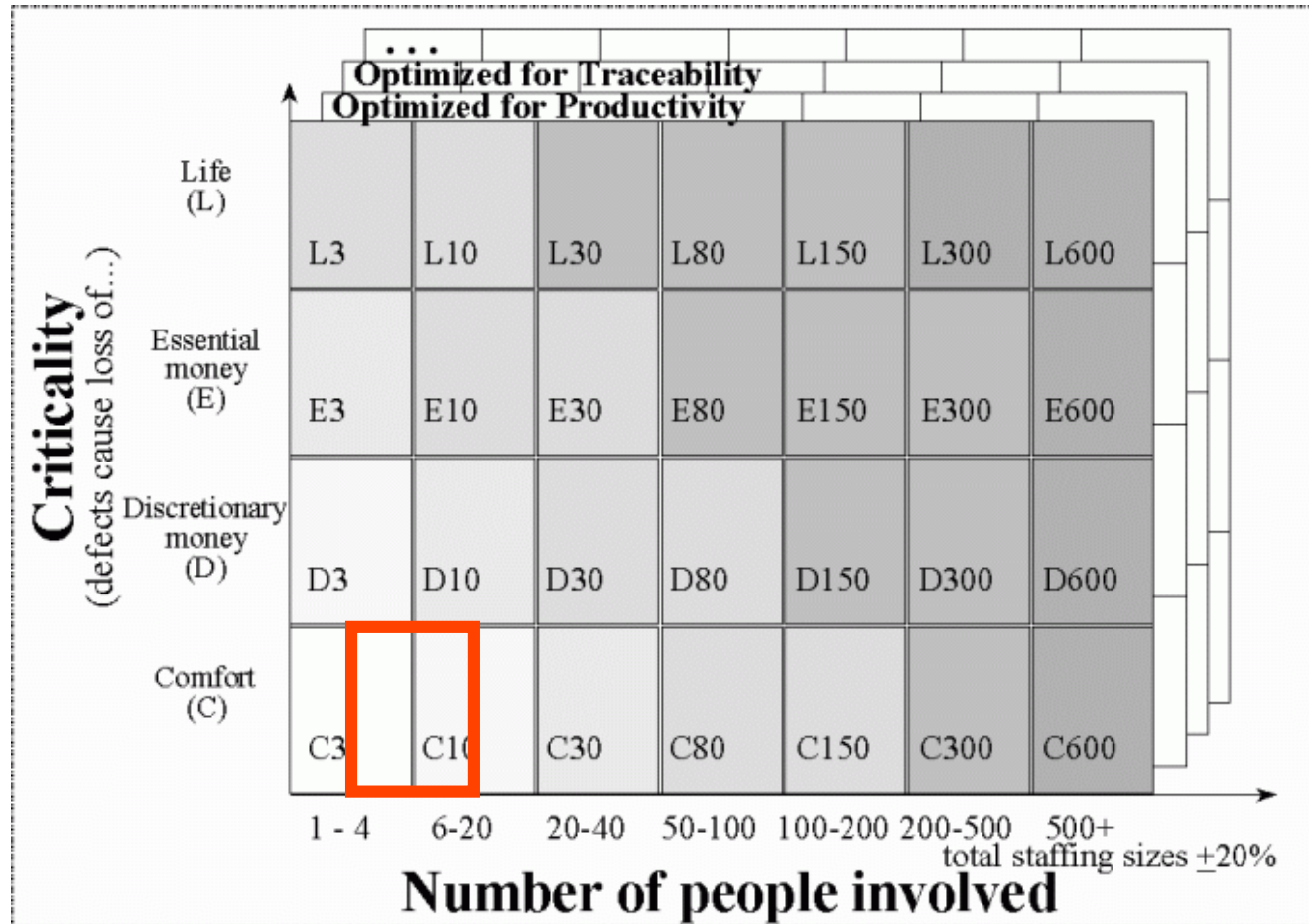
Исследовательские задачи

Дешево

Простота изменений

Маленькая команда

Например



Практики

Планирование крупными мазками

Трекер не нужен

Одна ветка в VCS

Каждый немножко аналитик

Практики

Планирование крупными мазками

Трекер не нужен

Одна ветка в VCS

Каждый немножко аналитик

У нас Agile

Запуск



Запуск

Много неожиданных требований

Много срочных требований

Очень частые обновление продукта

Объемы задач непредсказуемы

Запуск

Много неожиданных требований

Time-to-market

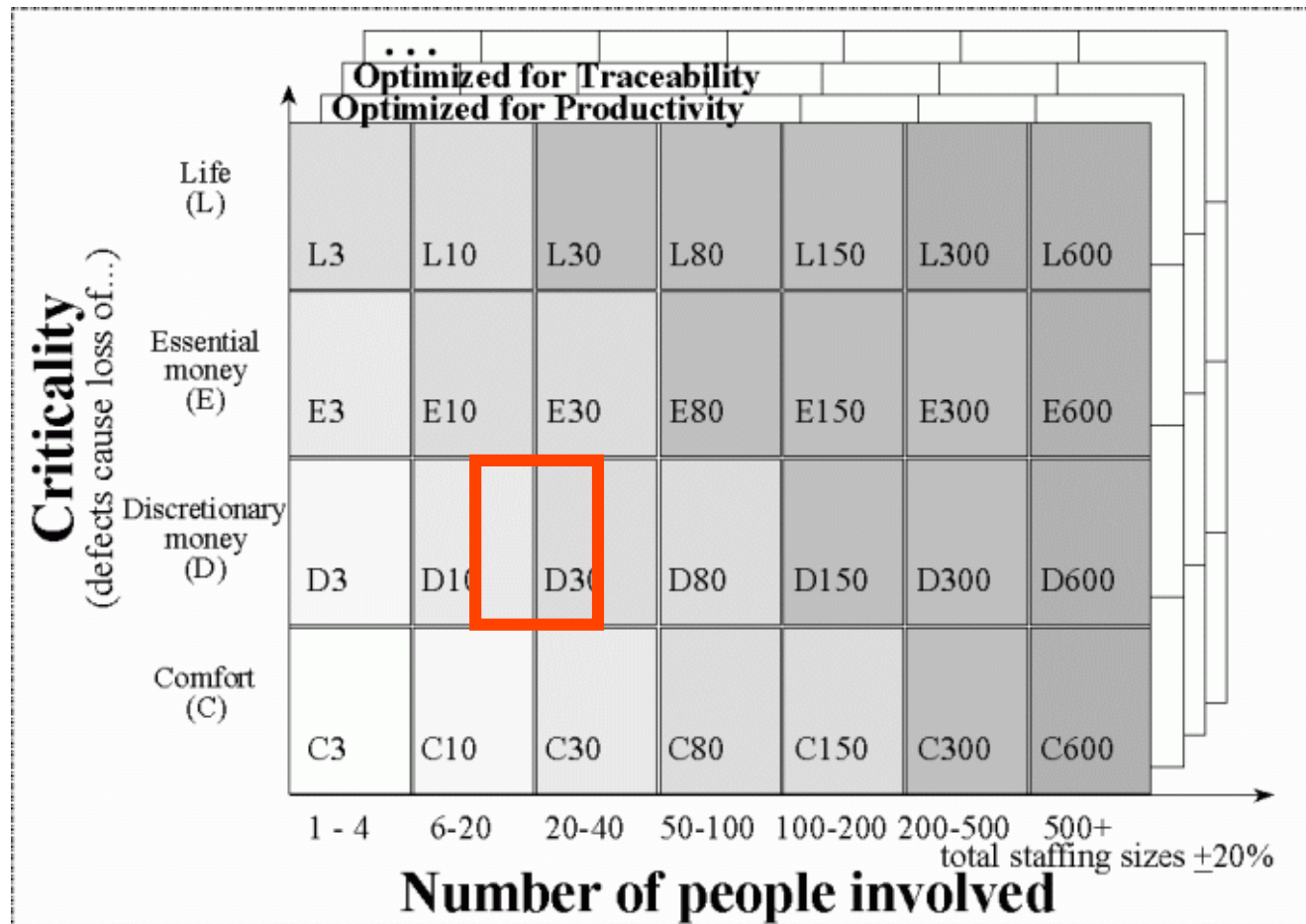
Много срочных требований

Надежность

Очень частые обновление продукта

Объемы задач непредсказуемы

Например



Практики

Чеклисты

Регулярность выкладок

Автоматизация тестирования

Мотивация

Практики

Чеклисты

Регулярность выкладок

Автоматизация тестирования

Мотивация

Однодневный SCRUM

Развитие



Развитие

Объемы задач предсказуемы

Регулярные выкладки

Важно ничего не испортить

Развитие

Объемы задач предсказуемы

Регулярные выкладки

Важно ничего не испортить

Предсказуемость

Надежность

Развитие

Объемы задач предсказуемы

Регулярные выкладки

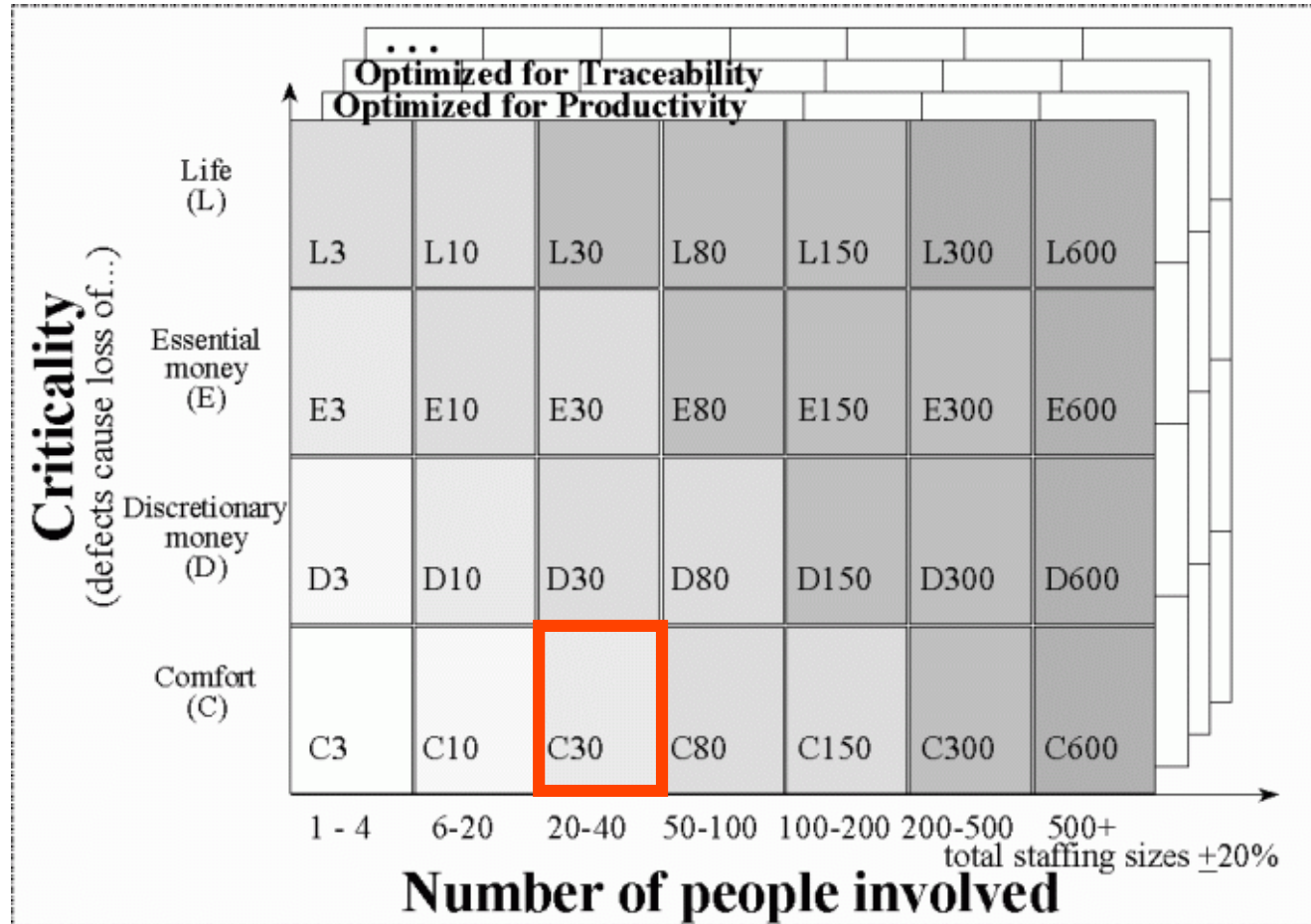
Важно ничего не испортить

Предсказуемость

Надежность

SCRUM? KANBAN?

Развитие



Как менять методологию

И не убить проект

Когда менять методологию?



Когда менять методологию?

Если понимаете «зачем?»

Когда менять методологию?

Если понимаете «зачем?»

Если продумали «как?»

Когда менять методологию?

Если понимаете «зачем?»

Если продумали «как?»

Если устраивает «почём?»

Как менять методологию

Анализ

Внедрение

Сопровождение

Анализ

Анализ

Описываем use cases

Решаем проблемы всех участников процесса

Определяем метрики успеха

Анализ

Описываем use cases

Решаем проблемы всех участников процесса

Определяем метрики успеха

Ищем инструментарий

Неудобным не будут пользоваться

Инструмент определяет возможности

Анализ

Описываем use cases

Решаем проблемы всех участников процесса

Определяем метрики успеха

Ищем инструментарий

Неудобным не будут пользоваться

Инструмент определяет возможности

Активно обсуждаем

Иначе что-нибудь забудем

Или кого-нибудь обидим

Внедрение

Внедрение

Уважаем своих коллег
Бережем чужое время
Помогаем при переходе

Внедрение

Уважаем своих коллег

Бережем чужое время

Помогаем при переходе

Пишем документацию заранее

С примерами решения постоянных задач

Никаких абстрактных flow, только конкретные примеры

Внедрение

Уважаем своих коллег

Бережем чужое время

Помогаем при переходе

Пишем документацию заранее

С примерами решения постоянных задач

Никаких абстрактных flow, только конкретные примеры

KISS

Пусть тропинки протаптываются сами

Сложные процессы редко работают

Важна простота изменений

Сопровождение

Сопровождение

Постоянно улучшаем

Собираем обратную связь

Как можно быстрее исправляем проблемы

Дописываем документацию

Сопровождение

Постоянно улучшаем

- Собираем обратную связь

- Как можно быстрее исправляем проблемы

- Дописываем документацию

Проверяем результат

- Помним изначальные цели

- Изучаем метрики

- Смотрим на реальное поведение

Практики

Некоторые полезные практики

Практики

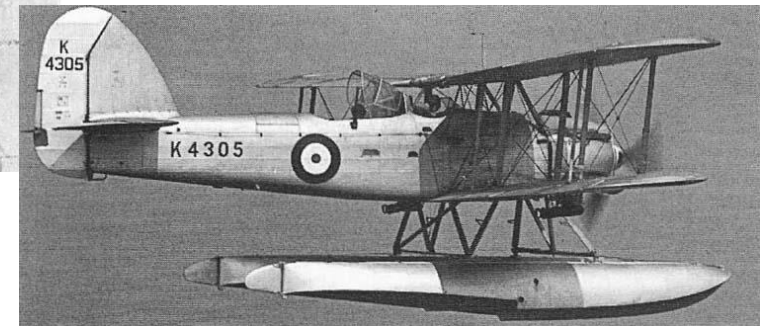
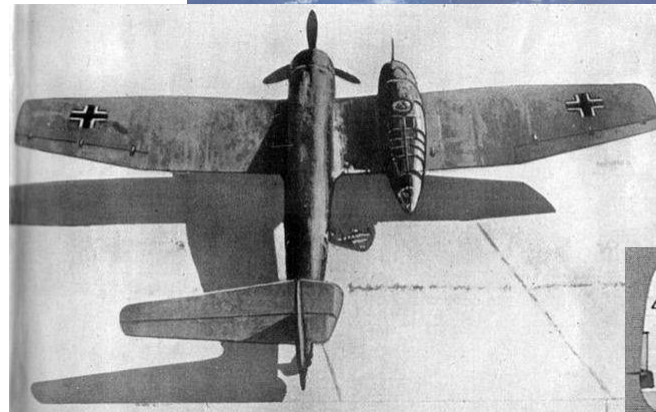
Право на «Зачем?»



Практики

Право на «Зачем?»

До трех не обобщать



Практики

Право на «Зачем?»

До трех не обобщать

Review before code



Практики

Право на «Зачем?»

До трех не обобщать

Review before code

IDE Driven Development

```

package Life

class Field(val width: Int, val height: Int, init: (Int, Int) -> Boolean) {
    private val Live: Array<Array<Boolean>> = Array(height) { i -> Array(width) { j -> init(i, j) } }

    private fun liveCount(i: Int, j: Int) = if (i in 0..height - 1 && j in 0..width - 1 && Live[i][j]) 1 else 0

    fun liveNeighbors(i: Int, j: Int) =
        liveCount(i - 1, j - 1) + liveCount(i - 1, j) +
        liveCount(i - 1, j + 1) + liveCount(i, j - 1) +
        liveCount(i, j + 1) + liveCount(i + 1, j - 1) +
        liveCount(i + 1, j) + liveCount(i + 1, j + 1)

    operator fun get(i: Int, j: Int) = Live[i][j]

    fun runGameOfLife(fieldText: String, steps: Int) { fieldText: "***" steps: 3
        var field = makeField(fieldText) field: Life.Field@530 fieldText: "***"
        for (step in 1..steps) { step: 1 steps: 3
            println("Step: $step") step: 1
            for (i in 0..field.height - 1) {
                for (j in 0..field.width - 1) {
                    print(if (field[i, j]) "*" else " ")
                }
                println("")
            }
            field = next(field)
        }
    }

    fun next(field: Field): Field {
        return Field(field.width, field.height) { i, j ->
            val n = field.liveNeighbors(i, j)
            if (field[i, j])
                n in 2..3
            else n == 3
        }
    }

    fun makeField(s: String): Field {
        val lines = s.replace(" ", "").split('\n').filter({ it.isNotEmpty() })
        val longestLine = lines.toList().maxBy { it.length } ?: ""
    }
}

```

Итого

Итого

Проекты бывают разные

Итого

Проекты бывают разные

Стадии проектов тоже разные

Итого

Проекты бывают разные

Стадии проектов тоже разные

Люди тоже разные

Итого

Проекты бывают разные

Стадии проектов тоже разные

Люди тоже разные

Всем нужны свои методологии!

Итого

Смена методологии – тоже проект

Итого

Смена методологии – тоже проект

Люди важнее процессов

Вопросы?

Дельгадо Филипп

dph.main@gmail.com

vk.com/dphil

