

Лабораторная работа №5

“Технический долг. Устранение долга”

Что такое технический долг:

<http://blog.crisp.se/2013/07/12/henrikkniberg/the-solution-to-technical-debt>

<http://blog.crisp.se/2013/10/11/henrikkniberg/good-and-bad-technical-debt>

ATDD для его устранения:

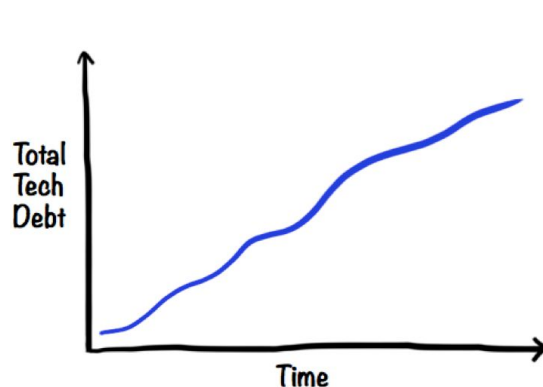
<https://www.infoq.com/articles/atdd-from-the-trenches>

Признаки технического долга на проекте:

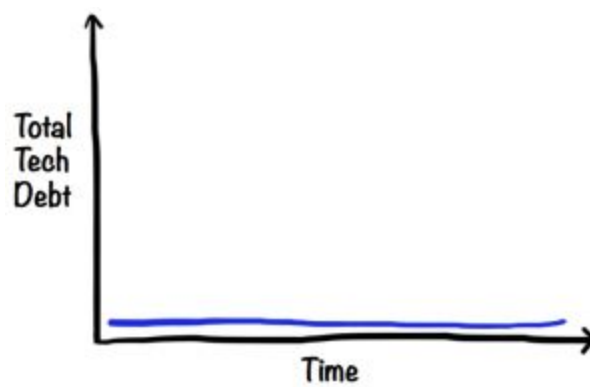
- непонятный / нечитабельный код
- дублирующийся код
- отсутствие автоматизации (тестов, сборки, развёртывания)
- запутанная архитектура и ненужные сложные зависимости
- медленные / неэффективные средства
- незакоммиченый код / долгоживущие ветки
- отсутствие / несоответствие технической документации
- отсутствие тестовой среды
- длинные циклы интеграции / отсутствие непрерывной интеграции

Графически:

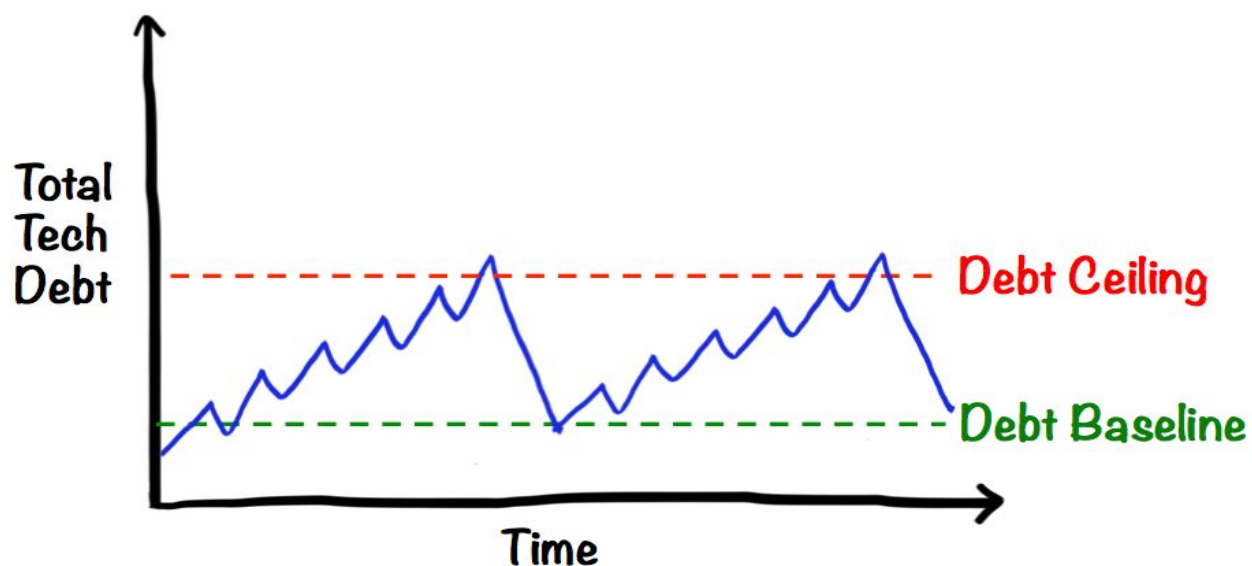
Как бывает:



Как мы хотим:

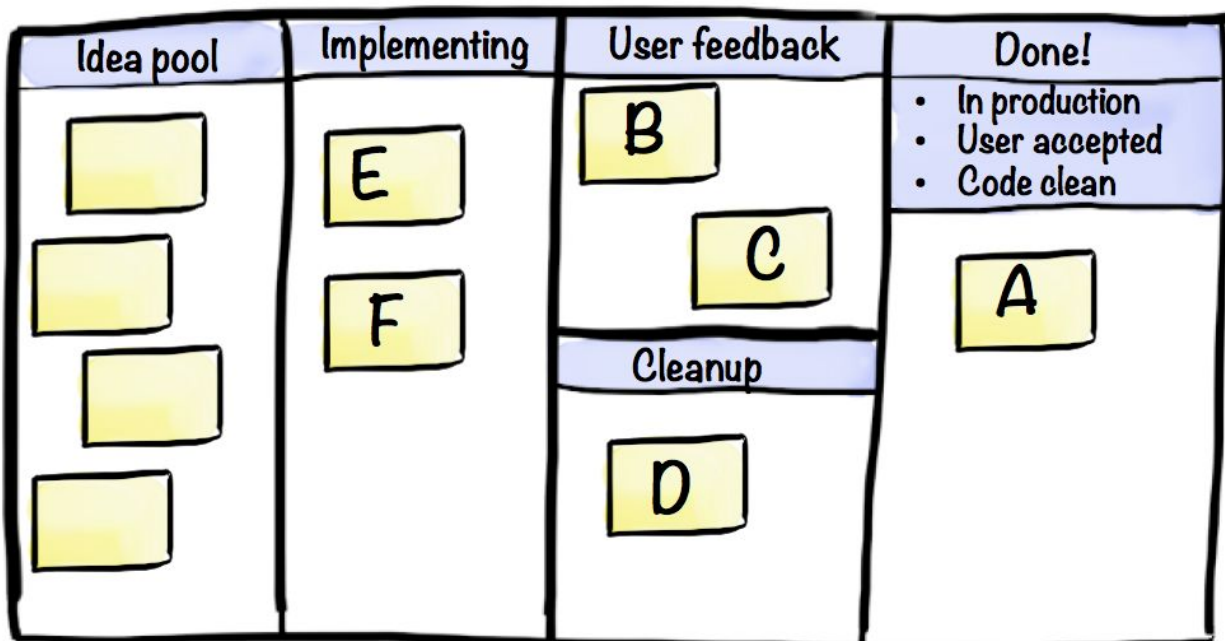


На самом деле хорошо, когда есть допустимый максимум и естественный минимум:



Один из способов - введение общих критериев приёмки (Definition of Done):

- Код чистый
- В продакшене
- Принято пользователем (протестировано на пользовательских сценариях - пункт "Как продемонстрировать" в описании User Story)



На рисунке:

Feature A: Всё сделано, задеплоено на продакшн, сценарии проходят

Feature B & C: В процессе прохождения сценариев

Feature D: Сценарии пройдены, проводим рефакторинг

Feature E & F: В процессе реализации

Задание на работу:

- Проинспектировать проект на наличие технического долга (привести по каждому из пунктов “Признаков”)
- Разработать план мероприятий по его устранению
- Оценить план мероприятий
- Сравнить объём долга и недоимплементированных фич
- Сделать выводы и обосновать необходимость (отсутствие таковой, частичную необходимость) устранения долга, внести коррективы в план разработки проекта
- Представить на проверку краткий отчёт, изменения плана.