

МОДЕЛЬ ЖИЗНЕННОГО ЦИКЛА  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ  
BUILD AND FIX

---

# ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Под моделью ЖЦ ПО понимается структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач на протяжении ЖЦ. Модель ЖЦ зависит от специфики, масштаба и сложности проекта и специфики условий, в которых система создается и функционирует.

Жизненный цикл программного обеспечения определяется как период времени, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации.

# CODE-AND-FIX ИЛИ BUILD-AND-FIX, КОДИРУЙ И ФИКСИРУЙ



**Модель Build-and-fix** – это модель неполного ЖЦ, которая пригодна для малых проектов ( $\approx 1000$  строк) и абсолютно непригодна для больших и сложных проектов с большим потенциалом развития.

По сути дела речь идет о модели проб и ошибок. Здесь, как мы видим на слайде, приходится строить продукт заново каждый раз до тех пор, пока клиент не будет удовлетворен. Этот цикл «модифицировать до удовлетворения клиента» и представляет собой наибольшую сложность этой модели.

Дело в том, что при разработке программного обеспечения в рамках данной модели, первичным условием является разработка детальных спецификаций, и реализация программного продукта без :

- существенной концептуальной проработки решения
- архитектурного проектирования,
- первичного проектирования.

То есть нам нужно повторить полный цикл разработки до тех пор, пока не будет получен тот продукт, который нужен заказчику. Тогда этот продукт, вместе с документацией, которая в данном случае тоже является достаточно сокращенной и обрывочной, передается заказчику и наступает стадия сопровождения.

Существенная сложность и проблема данного подхода — полный цикл разработки это достаточно дорогостоящее мероприятие, может получиться так, что те затраты, которые мы понесем в итоге, не окупятся за счет той экономии на первичных стадиях, анализ, первичное проектирование, архитектурное проектирование, которое нам придется из этого жизненного цикла в данном случае выбросить.