

# Задание 1. PDCA-цикл

Рассмотрим цикл на примере игровой студии, разрабатывающей игровой мультиплеерный проект

**Plan** (планирование): *что нужно сделать, когда это нужно сделать, кто должен это сделать, как это следует сделать и с помощью чего*

На этом первом шаге команда придумывает концепцию игры, и проводит начальную проработку игрового дизайна. Главная цель данного этапа – это геймдизайнерская документация, включающая в себя Vision (развернутый документ, описывающий игру, как конечный бизнес-продукт) и Concept Document (начальную проработку всех аспектов игры).

В продуктовой документации геймдизайнер формулирует и сохраняет свои идеи. Исполнителю документация позволяет правильно понимать свои задачи по реализации продукта. Тестировщик четко видит, что и как тестировать. Для Продюсера/ПМа эта документация предоставляет материал для формирования планов и контроля выполнения задач. Инвестор же (особенно на ранних этапах) получает понимание, на что именно он выделяет средства.

**Do** (действие): *выполнение запланированных работ*

Важный этап проектирования любой игры – это создание прототипа. То, что хорошо выглядит «на бумаге», совершенно не обязательно будет интересно в реальности. Прототип реализуется для оценки основного игрового процесса, проверки различных гипотез, проведения тестов игровых механик, для проверки ключевых технических моментов.

Очень важно на этапе создания прототипа реализовывать только то, что нужно проверить и в сжатые сроки. Прототип должен быть простым в реализации, т.к. после достижения поставленных перед ним целей, он должен быть «выкинут». Серьёзная ошибка начинающих разработчиков – нести временную инфраструктуру и «костыли» реализации кода в основной проект.

Цель данного этапа – получить минимально возможную полноценную версию игры, включающую в себя полностью реализованный основной игровой процесс. При этом высокое качество проработки обязательно нужно воплотить только для тех игровых элементов, которые существенно влияют на восприятие продукта. При этом все базовые фишки игры присутствуют как минимум в черновом качестве. Реализован минимальный, но достаточный для воплощения полноценного игрового процесса набор контента (один уровень или одна локация).

На следующем этапе производится достаточное количество контента для первого запуска на внешнюю аудиторию. Реализуются все фишки, запланированные к закрытому бета-тестированию. Задействуется наибольшее количество специалистов, которые занимаются производством всего основного наполнения игры. Художники создают все графические ресурсы, геймдизайнеры настраивают баланс и заполняют конфиги, программисты реализуют и полируют все фишки.

(Следует отметить, что действия, описанные в блоках Plan и Do также цикличны, т.к. разработчики, как правило, в процессе сталкиваются с непредвиденными трудностями и

изменении некоторых концепций и идей игры, в результате чего происходит перепланировка и действия согласно ей)

***Check (проверка):** сбор информации, выявление и анализ отклонений, установление причин отклонений*

Далее продукт впервые демонстрируется достаточно широкой публике, хотя и лояльной продукту или компании. Среди наиболее важных задач на этом этапе выступают: поиск и исправление гейм-дизайнерских ошибок, проблем игровой логики и устранение критических багов. На этом этапе в игре присутствуют уже все ключевые фишки, создано достаточно контента для полноценной игры продолжительное время, настроены сбор и анализ статистики. Тестирование идет по тест-плану, проводятся стресс-тесты уже с привлечением реальных игроков.

Далее продолжается тестирование игры, но уже на широкой аудитории. Идет оптимизация под большие нагрузки. Игра должна быть готова для приема большого трафика. В игре реализован биллинг и принимаются платежи.

***Act (корректировка):** корректировка планов с учётом информации, полученной на этапе проверки, проведение необходимых изменений*

На этом этапе полностью завершается разработка новых фишек. Происходит feature freeze, программисты перестают реализовывать что-то новое, а полностью переключаются на отладку и тюнинг имеющихся фишек. Геймдизайнеры, продюсер и аналитики делают выводы из собранной на этапе Check статистики и проверяют эффективность монетизации.

При этом, к началу этапа должна полностью функционировать инфраструктура проекта: сайт, группы соц. сетях, каналы привлечения (User Acquisition), поддержка пользователей.

(Здесь Check и Act также цикличны и происходят как бы параллельно, пока не будет произведён продукт, готовый к релизу)

На стадии релиза должно быть полностью отлажено оперирование продукта (техническая поддержка, работа с комьюнити), соблюдаются маркетинговые и финансовые планы, ведутся работы по улучшению финансовых показателей, активно отрабатываются каналы по привлечению трафика.

**----- Конец цикла -----**

Этот цикл может повториться либо для поддержки уже созданной игры (внедрение нового контента), или же для разработки новой игры, но уже с опытом разработки.

Пример, когда этот цикл неполный – это когда нет обратной связи от разработчиков или игроков (например, когда одни игнорируют других). Хороший пример – всякие третьесортные поделки на Unity в Steam-е (если интересно, посмотрите канал HappyWolf на Youtube, там полно видео про такие «игры», которые непонятно зачем существуют и непонятно, для кого сделаны). В этом случае мы имеем, как правило, сильно урезанные версии этапов Plan и Do, релиз, и на этом всё.

Преимущества PDCA, если он есть:

-Этот цикл позволяет повышать качество как уже созданного продукта, так и будущих продуктов

-Позволяет организации при создании новой фичи эффективно протестировать её, и только затем внедрять её в свой продукт. Это снижает риски и затраты при разработке

-Применимость алгоритма не ограничена, т.к его можно использовать не только для улучшения качества продукта, но и для упрощения самого процесса разработки.

Последствия отсутствия PDCA:

-Улучшение качества создания продуктов и модернизация уже существующих крайне маловероятна.

-Созданный продукт может оказаться попросту никому не нужным

-Процесс разработки будет гораздо более сложным, чем если бы PDCA применялся.

## Задание 2. Системные архетипы

### УРАВНОВЕШИВАНИЕ С ЗАДЕРЖКОЙ

Структура:



**Описание.** Человек, группа или организация, продвигаясь к цели, изменяют поведение в ответ на запаздывающие сигналы механизма обратной связи. Если они не учитывают запаздывания сигналов, то начинают использовать более сильные корректирующие действия, чем требуется, либо (порой) вообще отказываются от корректировок, поскольку не видят улучшения ситуации.

**Ранние симптомы.** “Мы думали, что достигли равновесия, но затем малость перегнули палку”. (Позднее возможен перегиб в обратном направлении.)

**Принцип управления.** При большой задержке энергичная политика порождает нестабильность. Нужно сохранять терпение либо сделать систему более чуткой к воздействиям.

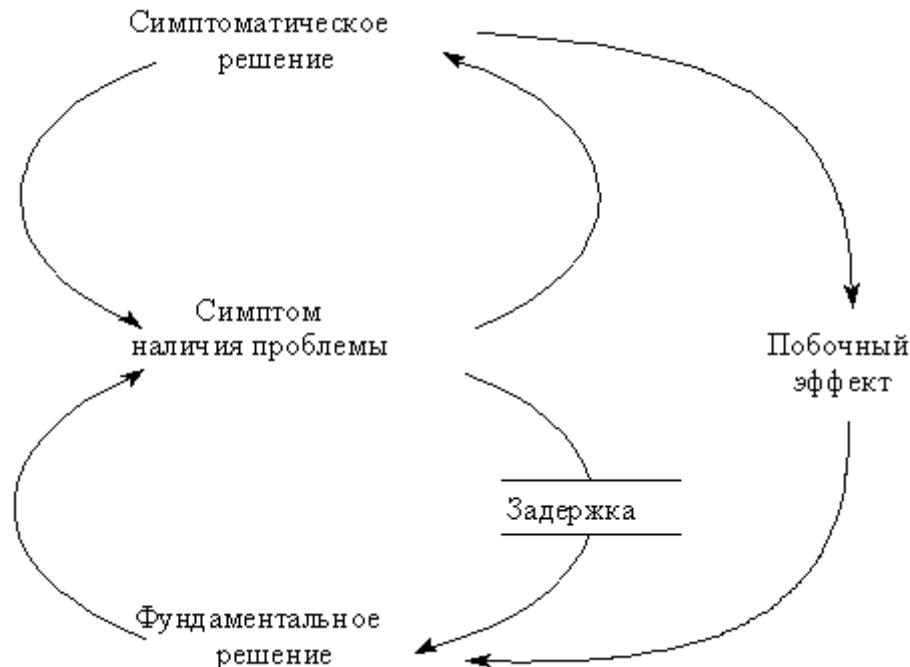
**Пример:** При использовании программы (например, браузера) компьютер начинает сильно лагать и не отвечать. Пользователь, в ответ на это, в злости начинает часто кликать по разным местам интерфейса программы, чем вызывает дополнительные события, которые ещё больше нагружают компьютер.

**Пример:** Разработка программного продукта командой из нескольких человек, коммуникация между которыми не налажена должным образом. Если разработка

последовательная (т.е. один делает одну часть, потом другой – другую, и т.д.), это ведёт к потере времени и каждому следующему человеку в этой цепи нужно это как-то корректировать.

## ПОДМЕНИТЬ ПРОБЛЕМУ

Структура:



**Описание.** Для корректировки ситуации используют краткосрочное “решение”, быстро дающее явно положительные результаты. Чем активнее и чаще прибегают к этому “решению”, тем реже применяют более кардинальные долговременные решения. Со временем способность использовать кардинальные меры может атрофироваться, что ведет к еще большей зависимости от симптоматических решений.

**Ранние симптомы.** “Смотри-ка, это решение сработало! Ты думаешь, нас еще ждут неприятности?”

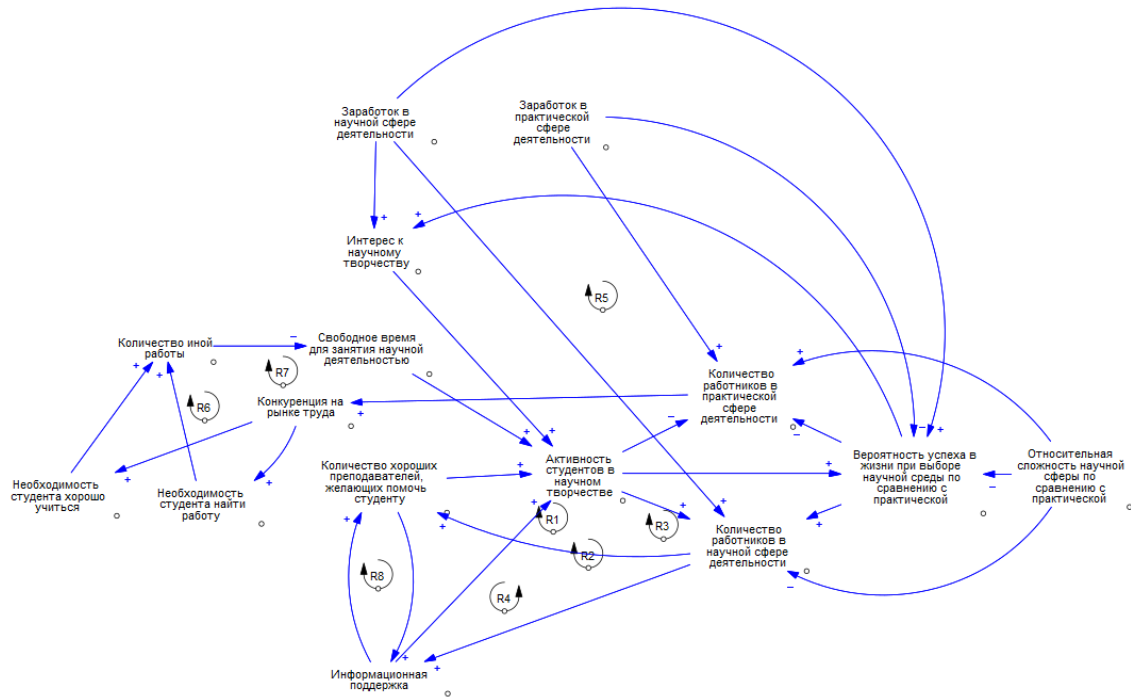
**Принцип управления.** Нужно сосредоточиться на кардинальном решении. Если без симптоматических мер не обойтись (время не терпит), их нужно использовать, чтобы выиграть время и дождаться результатов кардинальных решений.

**Пример:** При разработке игры какая-то механика не работает, как задумывалось (например, движок игры не позволяет реализовать её должным образом), но сроки поджимают, и разработчики прибегают к «костыльному» решению, которое позже приводит к багу и/или глитчу, который приходится потом фиксировать патчами после релиза.

**Пример:** Студент решил сделать лабу по теме, в которой он почти не разбирается. Ему надо сделать программу на Java. Делать её с нуля самостоятельно – долго и сложно, поэтому студент решает взять код однокурсника, изменить его и выдать за свой. Разумеется, на сдаче лабы преподаватель будет задавать вопросы по коду, и тут уже всё зависит от студента – либо он потратит достаточно времени на то, чтобы полностью разобраться в коде, и тогда спокойно сдаст лабу, либо начнёт плавать при ответе, и преподаватель отправит его на пересдачу (побочный эффект).

# Задание 3. Каузальная диаграмма (CLD)

Студенты практически не занимаются исследовательской работой в стенах университета (и не стремятся это делать). Почему так происходит?



Итак, начнём объяснение

Когда студент учится, ему необходимо хорошо учиться и необходимо найти работу, чем больше эти необходимости, тем больше работы у него в целом (практической деятельности). Это ведёт к тому, что у него меньше свободного времени (которое можно было бы посвятить научной деятельности). А это, в частности, ведёт к снижению активности студентов в научном творчестве.

Разумеется, это не единственный фактор, влияющий на эту активность. На это непосредственно влияют также такие факторы, как: интерес студентов к научному творчеству, количество хороших преподавателей, желающих помочь студенту, и информационная поддержка (сюда входят количество и доступность материала, необходимого для научной работы). Также, поскольку

Активность студентов в научном творчестве, очевидно, прямо пропорционально влияет на количество работников в научной сфере деятельности, и обратно пропорционально – на количество работников в практической сфере деятельности. Чем больше работников в научной сфере деятельности, тем больше хороших преподавателей, желающих помочь студенту и тем больше информационная поддержка. Чем больше работников в практической сфере деятельности, тем больше конкуренция на рынке труда, что повышает необходимость студента хорошо учиться и найти работу как можно скорее.

Также чем активнее студент будет в научном творчестве, тем больше вероятность его успеха при выборе научной среды по сравнению с практической, и эта вероятность прямо

пропорциональна количеству работников в научной сфере и обратно пропорциональна количеству работников в практической сфере.

Также на всё это оказывает влияние относительная сложность научной сферы по сравнению с практической. Она прямо пропорциональна количеству работников в практической сфере деятельности, обратно пропорциональна вероятности успеха в жизни при выборе научной среды по сравнению с практической

И ещё стоит упомянуть заработки в обеих сферах. Тут всё очевидно: чем больше заработок в научной, тем больше к ней интерес, количество работников в научной сфере и вероятность; чем больше заработок в практической, тем больше количество работников в практической и тем меньше вероятность.

Циклы:

R1 – R4: Количество хороших преподавателей, желающих помочь студенту -> Активность студентов в научном творчестве -> Количество работников в научной сфере деятельности -> Количество хороших преподавателей, желающих помочь студенту

Количество минусов 0 (чётное), поэтому это будет усиливающий цикл. Чем меньше преподавателей, тем меньше активность студента в научном творчестве. Чем меньше активность студента в научном творчестве, тем меньше работников в научной сфере. Чем меньше работников в научной сфере, тем меньше преподавателей. Также между активностью студентов и количеством работников можно вставить посредника в виде вероятности (да, это другой цикл, но так короче и проще объяснить). А ещё в цикле можно заменить количество преподавателей на информационную поддержку, т.к. эти понятия в целом смежные.

R5: Активность студентов в научном творчестве -> Вероятность успеха в жизни при выборе научной среды по сравнению с практической -> Интерес к научному творчеству -> Активность студентов в научном творчестве

Количество минусов 0 (чётное), поэтому это усиливающий цикл. Чем меньше активность студентов в научном творчестве, тем меньше вероятность успеха в жизни при выборе научной среды по сравнению с практической. Чем меньше вероятность успеха в жизни при выборе научной среды по сравнению с практической, тем меньше интерес к научному творчеству. Чем меньше интерес к научному творчеству, тем меньше активность студентов в научном творчестве.

R6 – R7: Активность студентов в научном творчестве --> Количество работников в практической сфере деятельности -> Конкуренция на рынке труда -> Необходимость студента найти работу или/и Необходимость студента хорошо учиться -> Количество иной работы (практическая деятельность) --> Свободное время для занятия научной деятельностью -> Активность студентов в научном творчестве.

Количество минусов 2 (чётное), поэтому это усиливающий цикл. Меньше активность -> больше работников -> больше конкуренция -> больше необходимость студента найти работу/хорошо учиться -> Больше количество иной работы -> меньше времени для занятия научной деятельностью -> меньше активность

R8: Количество хороших преподавателей, желающих помочь студенту -> Информационная поддержка -> Количество хороших преподавателей, желающих помочь студенту

Количество минусов 0 (чётное), это усиливающий цикл. Тут всё очевидно. Т.к. это смежные показатели, то они влияют друг на друга прямо пропорционально.

Возможные рычаги влияния:

1) Популяризация науки среди молодёжи.

Нужно дать понять молодёжи, что наука — это не сложно и не скучно. Любое занятие, превращенное в игру и приближенное к реальной жизни, перестает быть абстрактным, а значит — страшным. Геймификация образования приходит через проекты с виртуальной и дополненной реальностью, через компьютерные игры. Визуализация невидимой части мира, на уровне атомов или вирусов, позволяет представить себе, а что там внутри и почему оно так развивается. Также, в качестве метода преподнесения информации, можно использовать соцсети, блогеров и ютуб-каналы.

2) Улучшение информационной поддержки

В целом, чем лучше информационная поддержка, тем проще студентам осваивать науку, так что тут всё очевидно. Стоит уделить большое внимание доступности и лаконичности материала, меньше сложной теории, больше наглядной практики, примеров «из жизни»

3) Поддержка государством научной деятельности.

Как многие, наверное, знают, в нашем государстве (да и в других тоже) существуют научные фонды и организации-грантодатели, оказывающие финансовую поддержку научным, научно-техническим и инновационным проектам. Их работа оказывает весомую роль в формировании молодого научного поколения страны.

## Задание 4. Поиск коренной причины

<b>Описание контекста проблемы</b>	PLAN
Из-за долгого <u>релизного</u> цикла выручка задерживается и наша компания теряет клиентов. Из-за этого мы теряем клиентов, происходит задержка выручки и, как следствие, сама выручка становится меньше	
<b>Текущее состояние</b>	PLAN
У нас есть проблема: мы всегда срываем <u>дедлайны</u> . Точнее говоря, наши релизы всегда выходят позже запланированного времени. В среднем задержка составляет 20% от общего времени разработки, из-за неё мы теряем 20% клиентов и 20% доходов каждый релизный цикл (это значит, что если бы задержки не было, у нас было бы 100% клиентов и 100% доходов, но с задержкой мы имеем лишь 80% и тех, и тех).	
<b>Цель / Целевое состояние</b>	PLAN
Цель — осчастливить клиентов и получить <u>максимальную</u> выручку 1.25x ускорение поставки Устранение задержки выручки Минимизация потери клиентов 25% увеличение доходов (доходы могут увеличиваться как за счёт увеличения количества клиентов, так и за счёт уменьшения стоимости релиза)	
<b>Поиск корневых причин</b>	PLAN
<b>A3: Долгий релизный цикл</b>	
<b>Владелец:</b> Суражев Кирилл Леонидович	
<b>Ментор:</b> Форинко Борис Витальевич	
<b>Дата:</b> 06.12.2021	
<b>Контрмеры (эксперименты)</b>	DO
<b>1. Корневая причина</b> Недостаток автоматизации в процессе подготовки релизов <b>Контрмера</b> Автоматизировать процесс подготовки релизов. Это можно сделать с помощью инструментов автоматизации управления релизами (Application Release Automation — ARA). Как пример, можно внедрить автоматизацию тестирования на каждом этапе разработки <b>Ожидаемый результат</b> Стоимость релиза снизится, длина <u>релизного</u> цикла снизится	
<b>2. Корневая причина</b> Низкоприоритетные <u>фичи</u> не исключаются из релизов <b>Контрмера</b> Договориться с клиентом, что новые <u>фичи</u> могут быть добавлены в релиз только если из него исключаются то же число низкоприоритетных <u>фич</u> <b>Ожидаемый результат</b> Объём задач снизится, длина <u>релизного</u> цикла снизится	
<b>Проверка результатов</b>	CHECK
<b>1. =&gt; Стоимость и длина <u>релизного</u> цикла снизились</b> <b>2. =&gt; Объём задач и длина <u>релизного</u> цикла снизились</b> <b>Если обобщить, все основные цели были достигнуты:</b> - Задержка выручки была устранена - Поставка была ускорена в 1.25 раз - Доходы были увеличены в 1.25 раз	
<b>Дальнейшие действия (список задач)</b>	ACT
- Продолжить внедрять другие инструменты автоматизации управления релизами - Перенести приложения в облако (упрощает процесс управления релизами) - Использовать существующие решения для управления релизами	
<small>A3 Problem Solving Template v1.2 (April 2015) by <a href="#">Henrik Kniberg</a> and <a href="#">Tom Popp</a> License: <a href="#">Creative Commons Attribution 4.0 International</a> Original link: <a href="http://www.crisp.se/lean/a3-template">http://www.crisp.se/lean/a3-template</a> Перевод на русский: <a href="#">Дмитрий Лобачев</a> Большое примечание: <a href="http://lobachev.ru/a3-template">http://lobachev.ru/a3-template</a></small>	