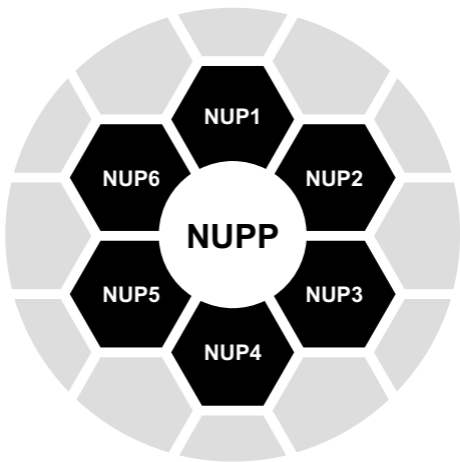


NUPP

Почти универсальные принципы проектов



Это загружаемая версия онлайн-руководства (<https://omimo.org/ru/modules/nupp/>), созданная 2026-07-02. Более новые версии доступны на сайте.

NUPP входит в семейство OMIMO (<https://omimo.org/ru/>), состоящее из открытых минималистичных модулей.

Это руководство можно свободно использовать и распространять в соответствии с лицензией Creative Commons Attribution 4.0 International.

Co-funded by the
European Union



OMIMO софинансируется Европейским союзом. Однако высказанные мнения и взгляды принадлежат исключительно OMIMO и не обязательно отражают позицию Европейского союза или EPOS

VZW. Ни Европейский союз, ни предоставивший финансирование орган не несут ответственности за их содержание.

Переведено: [Dmitry Ilienkov](#), [Valeriya Ilenkova](#), [Alexey Sokol](#)

Вступление

NUPP - это набор почти универсальных проектных принципов, которым лучше следовать во всех проектах, независимо от используемых методологий и подходов, для достижения максимального если мы хотим добиться успеха.

Каждый из известных подходов и методов для управления проектом опирается на некоторые из этих NUP (почти универсальные принципы). Однако необходимо учитывать следующие моменты:

- Обычно используются не все принципы, а для практиков было бы полезно рассмотреть все NUP, а не по отдельности.

- Основополагающие принципы обычно недостаточно разъяснены в подходах и методах, и большинство практиков настолько увлечены практическими деталями, что забывают о принципах и делают вещи, которые несовместимы с ними.

NUPP совместим со всеми основными методами, системами, подходами и фреймворками, такими как PRINCE2®, PMBOK® Guide, P3.express, PM², DSDM®, XP, and Scrum. Однако это может быть несовместимо с определенными интерпретациями этих систем, и именно здесь NUPP пытается побудить практиков пересмотреть свои интерпретации.

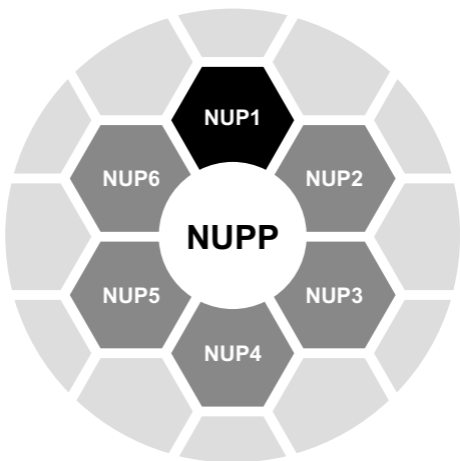
NUPP - это коллекция следующих NUP:

- [NUP1 — выбирай результаты и истину, а не привязанности](#)
- [NUP2 — береги и оптимизируй энергию и ресурсы](#)

- NUP3 — всегда будь проактивен
- NUP4 — помни, что прочность цепи определяется по самому слабому звену
- NUP5 — не делай ничего без четкой цели
- NUP6 — используй воспроизводимые элементы

NUP1

**выбирай результаты и
истину, а не
привязанности**



У всех нас есть естественное стремление принадлежать к группам, стремление, которое зачастую выходит за рамки изначальной формы, создает прочные связи и вызывает проблемы. Из-за привязанностей мы теряем намного больше, чем находим. Не ограничивая свою идентичность и взгляды привязанностями, мы можем добиться больше как профессионалы.

Пример: Agile vs Waterfall

Группа энтузиастов, достаточно смелых, чтобы попробовать адаптивные подходы в ИТ в то время, когда нормой было использовать предиктивный, собрались вместе и назвали свой подход «Agile». Это была отличная инициатива: не ограничивать свой выбор тем, что казалось обязательным. В Agile сообществе по-прежнему много энтузиастов и людей, ориентированных на результат, но, к сожалению, есть люди, превращающие Agile в культ и считающие

всех вокруг врагами. Это вызывает проблемы, например:

- Это не позволяет им учиться у кого-либо за пределами их группы
- Это мешает посторонним учиться у них
- Это делает принадлежность к группе важнее реальных целей, что, в свою очередь, мешает многим ее членам узнать истинное значение Agility.

Эту проблему можно существенно снизить или полностью убрать, если использовать термин «Agile» в отношении подхода к разработке, а не сообщества; и если люди, считающие себя создателями, решателями проблем и лидерами, будут рассматривать Agile как один из инструментов, а не как свою идентичность

Для настоящих профессионалов не существует войны между Agile и Waterfall.

Пример: PRINCE2® vs PMBOK®

Guide

В сообществе есть много профессионалов, которые связывают себя с PRINCE2® или PMBOK® Guide (обычно из-за своего географического положения) и не знакомы с другими подходами. У всех нас могут быть предпочтения в отношении определенных инструментов, но не стоит себя с ними идентифицировать - важнее ознакомиться со всеми, чтобы расширить наши горизонты и возможности выбора.

Настоящий профессионал открыт для всех идей, ищет их, изучает их и использует, когда требуется, вне зависимости от привязанностей.

Пример: непрерывное обучение

Привязанности удовлетворяют человека чувством принадлежности, которое они порождают, но не заставляют, а иногда даже мешают ему учиться из-за страха потерять это чувство. Если вы свободный

человек, эксперт без привязанностей, вам необходимо заполнить пробел обучением - непрерывным обучением.

То, что мы знаем сегодня, не является истиной в последней инстанции. Это только наше понимание истины в моменте, которое нужно улучшать, двигаясь вперед. Если ваши взгляды за несколько лет вообще не поменялись, что-то пошло не так. Это верно и в случае с NUPP: если вы вернетесь через несколько лет и увидите всё то же самое, у вас должно закрасться сомнение.

Пример: открытость

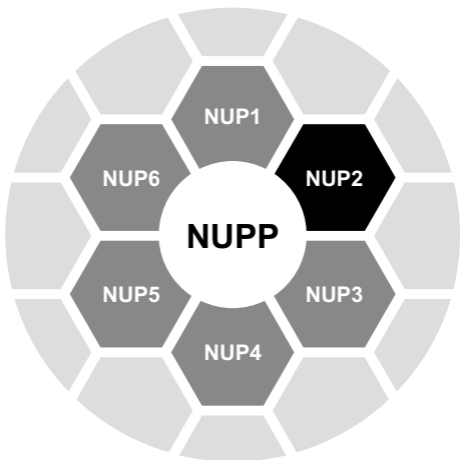
Возражая кому-то, убедитесь, что ваши возражения направлены на идею, а не на человека. Это поможет снизить напряженность. Аналогично, когда кто-то возражает вам или против вас, постарайтесь не принимать это на свой счет, сфокусируйтесь на обсуждении идей и оставайтесь открытыми для них. Не слушайте, чтобы ответить, а слушайте,

чтобы понять; работайте с другими над идеей, чтобы сделать ее лучше.

Некоторые люди могут преднамеренно нацеливаться на вас, а не на обсуждение идеи, и в этом случае вы должны помочь им сфокусироваться на ней, а не на вас, прежде чем продолжить. Постарайтесь сохранить эту сфокусированность в течение всего разговора.

NUP2

**береги и оптимизируй
энергию и ресурсы**



Ресурсы ограничены. Ресурсы, доступные для проекта, ограничены, так же как и

умственная энергия, необходимая для принятия правильных решений. Вы должны беречь и оптимизировать этот ресурс для себя и для проекта, и помогать другим членам команды делать то же самое.

Пример: правило 80/20

Большую часть возможных выгод от управления проектами можно получить, потратив небольшую часть усилий. В большинстве случаев стремление к получению 100% возможных выгод слишком дорого и неоправданно. Учитывайте это правило во всем, что делаете и побуждайте к этому других.

Пример: усталость от принятия решений

Мы используем один источник умственной энергии для принятия всех типов решений, а также для проявления силы воли. Если использовать его для

принятия ненужных или неважных решений, у вас будет меньше энергии для важных решений, что приведет к неудовлетворительным результатам. Это одна из причин, по которой лучше избегать микроменеджмента (принцип PRINCE2® «управление по исключениям»).

Конфликты, связанные с идеями, могут быть полезны, но конфликты, связанные с людьми, обычно вредны для проекта, и одно из последствий заключается в том, что это истощает умственную энергию членов команды. Если вы заметили такой конфликт, сделайте всё возможное, чтобы найти причину и устранить ее.

Пример: береги себя!

Решения, которые вы принимаете, и сила воли, которую вы выражаете, расходуют вашу умственную энергию. С другой стороны, энергия восполняется, когда вы спите и правильно питаетесь. Значит, вы должны хорошо заботиться о себе:

убедитесь, что у вас достаточно сна и отдыха, и хорошо питайтесь. Если у вас есть вредные привычки или проблемы со сном, не стоит бороться с ними в одиночку. Есть много специалистов, готовых помочь решить такие проблемы. Физическая активность также может помочь восполнить энергию, хотя исследования на тему не дают однозначных выводов.

Побуждайте членов команды делать то же, что и вы. Во-первых, убедитесь, что они работают в устойчивом темпе и без лишних переработок. Если у вас есть возможность, предлагайте питательную здоровую еду и напитки в рабочее время.

Пример: баланс между работой и личной жизнью

Многие из нас любят свою работу, но это еще не все, что нам нужно в жизни. Точно так же, как вы оптимизируете свою работу, вы должны планировать и реализовывать идеи в своей личной

жизни таким образом, чтобы она была радостной и счастливой. Когда вы счастливы в жизни, вы более успешны и на работе. Если можете, попытайтесь побудить членов вашей команды поступать так же.

Пример: мотивация

Мотивация – сложная концепция. Есть несколько интересных и полезных ресурсов по этой теме, в том числе и ненаучных. Тем не менее, полезно узнать больше о мотивации и стараться постоянно применять в работе, так как это увеличивает умственную энергию команды и возможность достижения лучших результатов для проекта.

Мотивация может принимать такие простые формы как улыбка и “спасибо” в знак признания работы. Обратите внимание, что некоторые из распространенных форм мотивации, например, небольшие денежные вознаграждения, могут дать

отрицательный эффект.

Пример: сотрудничество и командная работа

Сотрудничая, люди могут добиваться лучших результатов, но что еще важнее, мы по своей природе общительны и любим быть частью группы. Если вы сможете устранить негативные аспекты командной работы и создать благоприятную обстановку, ваша команда станет счастливее.

Вы должны быть осторожны, потому что люди разные, и некоторым нужно проводить больше времени в тишине и уединении - необходимо соблюдать баланс.

Пример: культура одной команды

Разные заинтересованные стороны имеют тенденцию группироваться, что вызывает напряженность в отношениях с другими

группами; например, люди, которые сосредоточены на бизнес-аспектах проекта vs люди, которые создают продукт. Это напряжение потребляет много энергии с обеих сторон, и это одна из причин, по которой мы должны попытаться создать культуру одной команды, где все работают вместе для достижения одной цели.

Пример: мудрость толпы

Когда большое количество людей с разным опытом и взглядами собираются вместе и работают в благоприятной среде, есть потенциал для получения очень хороших результатов, идей и решений, которые могут быть даже лучше, чем те, что приходят от отдельных экспертов.

Если у вас есть такая возможность, регулярно привлекайте членов команды к решению сложных вопросов в проекте. Помимо возможности получения хороших результатов, это также дает понять

членам команды, что их мнение ценится и что они играют важную роль в проекте, что, в свою очередь, повышает их уровень энергии. Шаг E02 из P3.express является примером использования мудрости толпы в проектах.

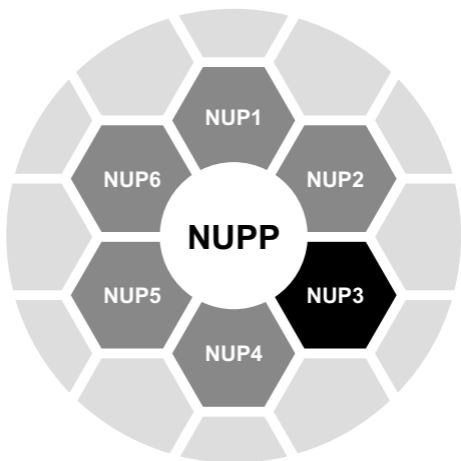
Пример: главный фасилитатор проекта

Если вы менеджер проекта, большая часть вашей работы имеет характер фасилитации (или, по крайней мере, должна иметь). С другой стороны, вы можете видеть, что члены команды в прошлом имели негативный опыт работы с менеджерами проектов, и что этот опыт влияет на их отношения с вами: вместо доверия часть их энергии расходуется на анализ вашего поведения на предмет потенциальных угроз. В таком случае вы можете изменить свою должность с руководителя проекта на главного фасилитатора проекта. В конце концов, это то, что вы действительно делаете в

проекте.

NUP3

всегда будь проактивен



Мы склонны быть реактивными, а не проактивными. Это может помочь нам сохранить нашу энергию при решении

второстепенных задач, или может дать нам лучшие результаты, когда мы имеем дело с чем-то, в чем мы совершенно некомпетентны. Эти ситуации отличаются от наших проектов, и здесь мы можем добиться лучших результатов, проявив инициативу.

Пример: планирование

Если вы решили куда-то поехать и уже опаздываете, можно выехать сразу, чтобы «сэкономить время», а возможные проблемы решать по мере их появления. Проактивный подход состоит в том, чтобы потратить некоторое время и сначала построить самый быстрый маршрут, учитывающий пробки, вероятные аварии и перекрытия, а только затем ехать; это займёт время, но в итоге вы его сэкономите, избежав проблем.

В противоположность тому, что некоторые люди думают об Agile проектах, планирование необходимо всегда, и оно зависит только от типа и уровня

детализации. Планирование перед выполнением является проактивным подходом.

Запомните выражение: дайте мне шесть часов, чтобы срубить дерево, и первые четыре часа я буду точить топор.

Если это предиктивный проект, вы можете потратить 4 часа на заточку своего топора, потому что вы уверены, что собираетесь срубить дерево. В Agile проекте вы не уверены, собираетесь ли вы рубить дерево, собирать сломанные ветви, стричь газон, добывать уголь или делать что-то еще. Тем не менее, вам все равно нужно подготовиться к этим работам в целом (узнать, где ближайший магазин инструментов) и подготовиться к конкретным работам (заточить топор), когда вы выберете конкретное решение - это тоже планирование.

Пример: планирование планирования

Планирование того, как мы собираемся выполнить проект, является проактивным подходом. Такая проактивность может быть даже расширена путем планирования того, как мы будем планировать выполнение. Это план управления проектом из Руководства PMBOK®, стратегия управления PRINCE2® и подходы в DSDM®.

Пример: непрерывное планирование

Реальность редко соответствует тому, что мы запланировали, и это нормально, но мы должны постоянно адаптировать наши планы, чтобы они оставались реалистичными и практичными. Мы должны делать это сразу, как только требуется корректировка, а не когда мы уже столкнулись с проблемами. Это проактивный подход.

Пример: управление рисками

Вся концепция управления рисками основана на проактивности: сталкиваясь с неопределенностью, мы не ждем, что произойдет, чтобы затем отреагировать, а заранее думаем о возможностях и последствиях, рассматриваем меры реагирования и, вероятно, что-то предпринимаем до того, как риск реализуется.

Помните, что то, что мы делаем в проектах, серьезно, иногда это затрагивает чьи-то жизни.

Пример: определение ролей и обязанностей

Вы можете оставить членов проектной команды работать без четких ролей и обязанностей, и рано или поздно появится форма ролей и обязанностей, но это будет слишком дорого и, в конце концов, может не сработать. Проактивный подход состоит в том, чтобы определить роли и обязанности на ранней стадии и адаптировать их по мере необходимости.

Это облегчит работу для всех, и поможет сосредоточиться на создании продукта, вместо того, чтобы решать, кто чем занимается.

Количество и разнообразие ролей зависят от типа и размера проекта; их набор может быть строго определен, как, например, в Scrum, умеренно, например, в P3.express, или всеобъемлюще, как в DSDM® и PRINCE2®. Однако не забывайте, что описания ролей в этих методах касаются только управленческих аспектов, – вам всегда нужно добавлять описания ролей и для технических аспектов.

Пример: доступные варианты

Стоит ли преждевременно закрыть проект или продолжить?

Очень редко у нас действительно есть только два варианта, даже если вопрос подразумевает это. Вам нужно проявлять проактивный подход и обдумать все

варианты, прежде чем принимать решение. Может быть, вы можете изменить содержание проекта; возможно, вы можете приостановить его, пока что-то еще не станет ясным; или, может быть, вы можете изменить подход к проекту (например, отдать его на аутсорсинг) и т. д.

Пример: критическое мышление

У всех нас много предубеждений, которые, с одной стороны, помогают нам выживать, а с другой – приводят к некачественным решениям. Принимая важные решения по проекту, лучше всего взять паузу и рассмотреть все предубеждения, которые могут повлиять на наше решение, прежде чем они вызовут проблемы.

В качестве ссылки вы можете использовать список когнитивных искажений, приведенный в [Википедии](#).

Существуют даже специальные

фреймворки, которые вы можете использовать для получения качественных решений. Поначалу их использование может отвлекать и даже раздражать, но вскоре вы привыкаете к ним и получаете преимущество от их применения без особых сознательных усилий.

Пример: прозрачность

Нам не нравится срывать сроки или сталкиваться с другими проблемами, но это не значит, что их нужно скрывать. Вы должны сохранять прозрачность и открыто сообщать о проблемах стейкхолдерам, потому что некоторые из них могут вам помочь, и, кроме того, рано или поздно они в любом случае узнают о проблемах и их последствиях, а от некоторых из них могут потребоваться скорейшие действия (например, принять негативные последствия).

Пример: общаться эффективно

Бывает, что вы отправляете отчеты и не получаете обратной связи. Вы можете подумать, что все в порядке только потому, что отрицательных отзывов нет, хотя это может быть и не так. Вы должны быть проактивны и проверять, ознакомились ли они с отчетом и послужил ли он своей цели, и использовать эту информацию для совершенствования коммуникаций; в противном случае, эта скрытая проблема может вызвать серьезные проблемы позже, когда её будет слишком трудно исправить.

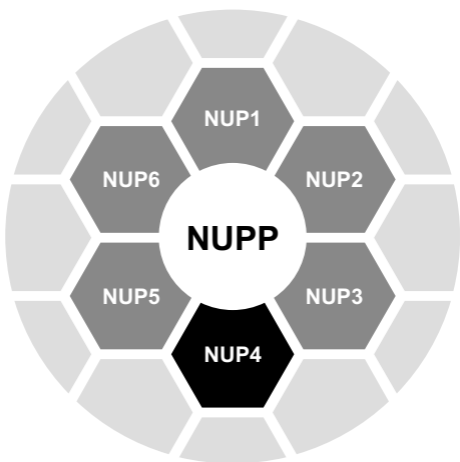
Пример: берите на себя ответственность

Легко обвинять других в плохих результатах. Например, вы хотите, чтобы ваша организация предоставила вам карт-бланш, и чтобы вы сделали проект идеально, но они этого не делают, и в результате проект терпит неудачу. Это не проактивный подход.

Проактивный подход заключается в том, чтобы взять на себя ответственность и делать все возможное в рамках существующих ограничений. Не стоит ожидать, что организация полностью вам доверяет и даст зеленый свет в надежде на лучшие результаты, особенно если они уже видели неудачные проекты. Сделайте одно небольшое улучшение в рамках существующих ограничений и используйте это, чтобы завоевать доверие, получить немного больше ресурсов и терпения, а затем используйте это для чуть большего улучшения и так до тех пор, пока не достигнете своей цели.

NUP4

**помни, что прочность
цепи определяется по
самому слабому звену**



Проекты состоят из различных доменов, и все они требуют внимания; у нас должна быть целостная картина проекта.

Недостаточно обратить внимание на, казалось бы, самый важный домен (например, сроки), потому что все домены взаимосвязаны и не работают должным образом без достаточного внимания.

Пример: это всё о дедлайне!

Допустим, вы строите что-то для Олимпийских игр. У проекта очень строгий дедлайн, что делает управление сроками очень важным. Так ли это?

Что, если качество настолько низкое, что через некоторое время требуется повторная работа. Это повлияет на время, значит, нужно уделять внимание качеству и срокам. У вас может быть причудливый сад, указанный в первоначальном определении проекта, но вы знаете, что если не хватает времени, вы можете не выращивать его и просто заменить газоном, если вы вовремя

рассмотрели эту возможность и подготовились к ней. Таким образом, управление содержанием также важно. Теперь содержание, время и качество в центре нашего внимания.

Вы слышали о знаменитом случае, когда президент Кеннеди встретил уборщика в NASA и спросил его, что он делает? Уборщик ответил: «Я помогаю посадить человека на Луну». Разве такие люди в проекте не помогают выполнить его в срок?

Продолжая, вы замечаете, что каждый отдельный домен в проекте способствует управлению сроками, и вы не можете уложиться в срок с приемлемым уровнем вероятности, если не уделяете внимание всем доменам.

Пример: выборочные заимствования

Когда люди сталкиваются со множеством методов, иногда они начинают делать

выборочные заимствования из разных систем и создают смесь всего, что кажется им интересным. Это обычно не работает, потому что элементы не работают изолированно и должны быть совместимы друг с другом. Любые дополнения или изменения в системе должны быть сделаны с комплексной точки зрения.

Вот почему мы иногда видим противоречивые элементы в разных подходах; что-то хорошо работает в одном, а его противоположность хорошо работает в другом. Сами по себе эти элементы не являются ни правильными, ни неправильными.

Самый безопасный подход - это выбрать методологию для проекта, адаптировать ее, а затем осторожно добавить к ней новые элементы, учитывая целостность всей системы.

Пример: антипроцессный подход

Одна из лучших вещей, которую сделали Agile-методы, – привлечение внимания к человеческим аспектам. Agile Manifesto дает большую ценность личностям и их взаимодействию по сравнению с процессами и инструментами, хотя это может и не быть справедливым сравнением. Почти все подходы признают важность человеческого аспекта, но именно в аджайл они встроены в процессы, а не являются просто рекомендацией. Таким образом, речь идет не о конкуренции между человеческими аспектами и процессами, а скорее о том, как они рассматриваются в системе.

Нет сомнений в том, что некоторые пытаются заменить человеческие аспекты усложненными процессами, но это всего лишь злоупотребление. Существует также и обратная проблема: люди пытаются заменить процессы человеческими аспектами, что также не очень хорошо работает.

Пример: вам нужны все домены

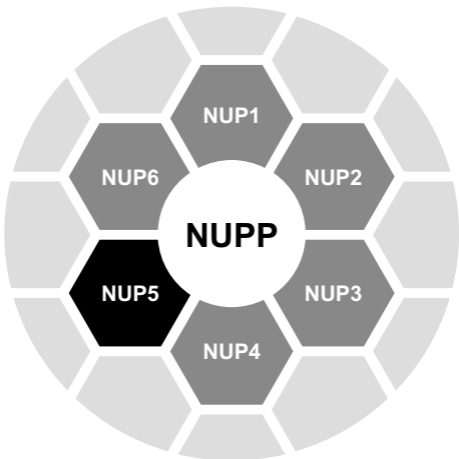
Когда вы думаете о доменах, будьте осторожны, чтобы не пропустить ни один из них. Кстати, что это за домены? Если вы проверите основополагающие ресурсы, вы получите разные ответы; и все же, ни один из них не является полной правдой.

Темы PRINCE2® - это домены, но это только те домены, которые играют ключевую роль в методологии. Другие домены только подразумеваются. Руководство PMBOK® не является методологией и может сформулировать ответ гораздо лучше с помощью десяти областей знаний. Однако это интерпретации всех доменов, основанные на точке зрения руководства PMBOK® на проект, а не на нейтральной (нейтральная точка зрения существует не всегда). Например, человеческие аспекты не получают большого внимания в Руководстве PMBOK®.

Хорошим источником информации о доменах является ISB. Однако здесь речь идет не о доменах, а о компетенциях, которые требуются в проекте. У них нет однозначных сопоставлений с доменами, но это очень помогает в их идентификации. В NUPP нет списка доменов, в первую очередь потому, что это скорее метасистема, чем система, а также потому, что категоризация доменов зависит от типа проекта и его среды; например, рутинный строительный проект может нуждаться в ином подходе, чем творческий исследовательский проект.

NUP5

**не делай ничего без
четкой цели**



Вы не должны ничего делать, если это не имеет четкой цели. Представьте себе два

параллельных мира, в которых все одинаково, за исключением того, что вы планируете делать: насколько разными будут эти миры? Разве разница стоит усилий, чтобы сделать это?

Если у вас нет ясной цели и вы делаете что-то только потому, что это делают все остальные, или все говорят, что это важно, то

- это может не принести вам реальной пользы, и, следовательно, вы можете ничего не получить от этого;
- или же в вашем случае он все еще может иметь потенциальные выгоды, но поскольку у вас нет этой цели, ваш способ сделать это может не помочь реализовать преимущества.

Пример: портфолио и программы

Если вы участвуете в отборе и запуске проектов, убедитесь, что вы

сосредоточены на решаемых проблемах и создаваемых преимуществах, а не на технических решениях. Классическим примером является производитель лифтов, который получая жалобы на скорость своих лифтов, запустил несколько проектов по увеличению скорости без особого успеха. Это продолжалось до тех пор, пока они не решили сосредоточиться на проблеме (скука или дискомфорт людей) вместо «естественного» решения (более быстрые лифты). В результате они добавили зеркала в лифты, и это решило проблему просто и дешево.

Помните, что управление проектами - это в основном правильные действия, а управление портфелями - правильные вещи. Неважно, насколько хорошо вы управляете своими проектами; это не сработает, если вы делаете неправильные проекты. Всё это о наличии цели.

Пример: проект в целом

Гибкость продукта варьируется между проектами. В некоторых проектах по развитию ИТ продукт является полностью гибким, и его окончательная форма зависит от обратной связи, которая будет генерироваться по инкрементам продукта в ходе проекта, что требует применения адаптивного (гибкого) подхода. На практике это комбинация уровней портфеля, программы и проекта и требует максимального внимания к конечной цели. Рекомендуется задокументировать цель и сделать ее доступной; это одна из целей «видения продукта», которое используется в некоторых проектах Scrum. Внимание к бизнес ценности в Agile проектах - это их способ обеспечить соответствие цели.

В других типах проектов, где продукт является относительно фиксированным и существуют другие механизмы, гарантирующие, что конечный продукт будет удовлетворять цели, члены

проектной команды могут перенести большую часть своего внимания с цели на продукт (следовательно, принцип «сосредоточить внимание на продуктах» (PRINCE2®)), но все же требуется по крайней мере минимальное внимание цели, чтобы гарантировать, что то, что строится, будет соответствовать цели, что можно сделать путем сравнения прогнозируемых выгод с ожидаемыми (принцип «постоянная оценка целесообразности» и тема «экономическое обоснование» в PRINCE2®).

Когда проект запускается для внешнего клиента, у клиента будет своя цель проекта, а у вас своя. Чтобы преуспеть, вам нужно понимать обе.

Пример: мониторинг проекта

Необходимо сосредоточиться на конечной цели, но она может быть слишком абстрактной для многих повседневных задач. Вот почему для эффективной

работы необходима иерархия целей. Сначала обоснование и выгоды проекта определяются на основе конечной цели, а затем у вас появляются явные и неявные цели для переменных проекта (например, сроки, стоимость и качество), достигая которые вы, в свою очередь, удовлетворяете конечную цель. Это цели более низкого уровня, которые полезны для многих наших повседневных задач.

Когда дело доходит до мониторинга, низкоуровневый мониторинг проекта будет осуществляться с использованием самого низкого уровня переменных, поскольку отследить конечную цель не всегда возможно. В этом случае держите в уме: какая цель мониторинга?

Нормальный и приемлемый ответ - посмотреть, находимся ли мы на правильном пути, и если нет, совершить определенные действия, чтобы вернуться в прежнее русло, или скорректировать цели и посмотреть, сможем ли мы по-прежнему достигнуть конечную цель.

Поэтому наши измерения должны быть в состоянии помочь справиться с этой низкоуровневой задачей, а также сформировать прогноз по завершении для переменных проекта; например, когда мы сможем завершить проект? Сколько денег нам понадобится, чтобы закончить проект?

Все остальные измерения, такие как плановые и фактические значения на сегодняшний день, являются лишь промежуточными значениями, которые вам нужны для расчета прогнозов, а не окончательными значениями, которые вы используете для управленческих целей.

Пример: документы

Независимо от того, какой подход к разработке вы используете в проекте, планирование всегда необходимо. Важным вопросом является уровень детализации каждого типа плана. Если в плане недостаточно подробностей, план не сможет внести достаточный вклад, и

выполнение проекта будет основано на спонтанных решениях, а не на комплексных, целостных решениях. С другой стороны, многие люди стараются быть осторожными и добавляют слишком много деталей к своему плану, что делает его слишком сложным для подготовки и сопровождения и слишком жестким для неопределенности проекта. В результате план становится нереальным и бесполезным.

Лучший способ определить уровень детализации - это иметь цель для каждого элемента в плане. Например, если вы рассматриваете возможность добавления ресурсов в план, у вас должна быть для этого цель: как вы собираетесь его использовать? Как это поможет проекту? Сколько усилий это займет? и стоит ли это того?

Иногда нужно решить, хотите ли вы иметь определенный элемент в своих планах, а иногда - то, как вы хотите спланировать или подготовить что-то. Будь то

экономическое обоснование, устав проекта или отчет: вы все равно должны спросить себя, почему вы готовите этот документ и как он может помочь проекту.

Поиск «шаблона» - это противоположность тому, чтобы делать что-то на основе цели.

Пример: статус-отчеты

По проектам часто формируют действительно длинные статус-отчеты. Основываясь на этом NUP, мы должны спросить себя, какова цель отчета и как мы можем достичь этой цели независимо от того, как поступает большинство других людей.

Это может привести к тому, что мы начнём формировать очень простые одностраничные отчеты, которые заинтересованные стороны действительно прочитают и поймут, в противоположность длинным. Это означает внимание к цели.

Однако когда вы готовите одностраничные отчеты, некоторые люди могут возражать и думать, что у вас нет «надлежащей» системы мониторинга. На практике это создает для вас вторую цель (помимо первой цели – помочь заинтересованным сторонам понять статус проекта), и чтобы удовлетворить её, вы можете просто создать второй тип отчета, который будет очень длинным. При этом вы не будете смешивать его с первым отчетом, потому что эти две цели не совпадают.

Пример: бизнес-кейс и устав проекта

Подготовка документов, таких как бизнес-кейс и устав проекта, обычно является скучной, бесплодной, бюрократической необходимостью, в то время как все эти документы имеют обоснование и применимы к большинству проектов. Если вы попытаетесь найти «шаблон» и заполнить его, работа будет просто

сделана впустую; тогда как вы можете вместо этого проверить реальное назначение этих документов, посмотреть, могут ли они помочь в вашем проекте и каким образом они могут помочь, а затем подготовить документ так, как вам нравится, просто для достижения этих целей: это будет правильный документ.

Пока вы думаете о том, как подготовить документы для достижения их целей, вы можете забыть рассмотреть все варианты. Чтобы избежать этого, вы можете проверить, что предлагают PRINCE2®, PMBOK® Guide, P3.express и DSDM®, а затем оценить эти рекомендации в контексте ваших целей.

Пример: пост-проект

Хотя проект имеет конкретную цель, и мы можем учитывать эту цель на протяжении всего проекта, реализация цели в основном оценивается на основе прогнозов, сделанных в ходе проекта. Однако не следует забывать об этом,

когда проект завершен. Важно проверить реализацию целей после завершения проекта, потому что

- мы можем увидеть, как достигаются конечные цели и извлечь из этого уроки для будущих проектов, и
- иногда цели достигаются только тогда, когда мы предпринимаем некоторые дополнительные действия после завершения проекта, и понимание необходимости этих дополнительных задач требует контроля.

Постпроектный мониторинг является необходимым шагом для достижения цели. В основном это зона ответственности управления программами и портфелями, и поскольку большинство компаний не имеют таких уровней управления, этим шагом обычно пренебрегают. Вот почему такие методы, как P3.express и DSDM®, добавляют этот шаг в свои методологии управления

проектами.

Пример: простота

Мир сложен и хаотичен, но наши модели представляют собой абстрактные приближения, которые отражают части мира, и, следовательно, они могут быть простыми. С другой стороны, простые системы обычно работают лучше, потому что мы можем сосредоточиться на основной идее.

Многие из нас пытаются добиться лучших результатов, добавляя сложность к нашим системам, надеясь, что она будет соответствовать сложности всего мира, хотя на практике это затрудняет работу с системой и, как правило, мешает нам достичь цели.

Пример: Адаптация

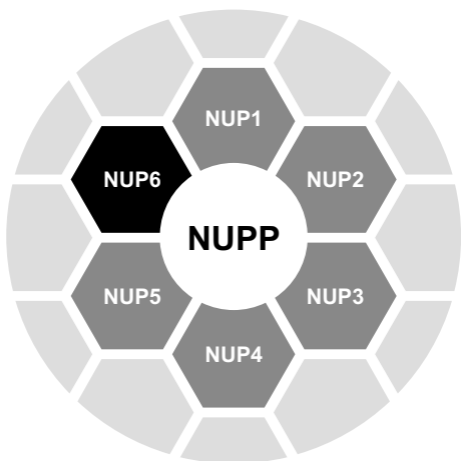
Программное обеспечение для потоковой музыки очень отличается от ПО, которое будет использоваться для оборудования в

больнице или самолете, где от него может зависеть жизнь людей, или от ПО, которое будет использоваться на спутнике, где оно, как предполагается, будет работать много лет без сбоев, всё это отличается от строительства летней дачи, пожарной станции или завода.

Фокусируясь на целях, вы лучше поймете, как адаптировать системы и артефакты для различных проектов.

NUP6

**используй
воспроизводимые
элементы**



Спонтанный подход к проекту требует слишком много энергии и ресурсов, и вы всегда рискуете пропустить некоторые из необходимых элементов. Лучший способ упростить работу - использовать воспроизводимые элементы, желательно в повторяющихся циклах.

Пример: чек-листы качества

Чек-лист — простой пример потенциально повторяемого элемента, который многие люди используют и в личной и профессиональной жизни. Возьмите, например, критерии качества продукта:

- Во-первых, вы можете создать список всех критериев, что уже является формой планирования.

- NUP6 рекомендует попытаться обобщить их: есть ли другие подобные продукты в проекте? В таком случае сделайте общий чек-лист качества для этой категории продуктов. Если возможны вариации, сохраните общий список и добавьте несколько дополнительных элементов для отдельных продуктов. Теперь у вас есть воспроизводимые чек-листы.

- После того, как вы подготовите общие чек-листы для различных типов продуктов, вы можете найти повторяющиеся пункты и сделать для них виртуальную родительскую категорию. В этом случае вместо повторения элементов для всех этих общих чек-листов вы можете извлечь их и поместить в родительский чек-лист. В конце у вас, вероятно, будет единый общий чек-лист для всего проекта. «Definition of Done» в Scrum является примером использования чек-листов на уровне проекта для проверки качества (возможно, среди прочего). Таким образом, каждый результат будет принадлежать иерархии категорий и должен удовлетворять элементам, которые появляются в контрольных списках всех категорий в их цепочке.

Таким образом, элемент в родительском чек-листе станет повторяемым для всех результатов, находящихся под ним, что

экономит время и энергию при планировании и выполнении.

Что еще более важно, как только вы сделаете это для одного проекта, вы можете адаптировать и использовать его для всех аналогичных проектов в будущем, что является повторяемой формой планирования для нескольких проектов.

Пример: процессы и рабочие процессы

Некоторые продукты или связанные с ними цели требуют стандартизации и воспроизводимости определенных шагов. Например, если для достижения результатов требуется индивидуальное проектирование и утверждение, можно подготовить простую последовательность шагов, которая четко определит все этапы, участников и примерное время, тем самым избегая многих трудностей. Однако следует соблюдать осторожность, чтобы не усложнять процессы и

процедуры, поскольку это будет иметь негативные последствия. Все участники проекта должны рассматривать бизнес-процессы как поддержку и упрощение, а не как бюрократию, мешающую работе.

Гибкие проекты имеют воспроизводимые элементы в своем итеративном подходе к разработке, где определенный тип действий повторяется для каждой фичи; например, обычная ежедневная рутина в XP (экстремальном программировании): объединение в пару, выбор элемента, проектирование на доске, создание сценариев теста и кода, интеграция кода и т. д.

Помимо воспроизводимых процессов, которые можно использовать для технических действий, у вас могут быть повторяемые элементы для действий по управлению проектами. Процессы в Руководстве PMBOK®, PRINCE2® и DSDM®, шаги в P3.express и события в Scrum являются примерами этой концепции.

Пример: циклы

Полезно использовать воспроизводимые элементы для управления проектом. Это может быть сделать еще проще, помещая их в повторяющиеся циклы. Эти циклы значительно упрощают повседневную работу людей, вовлеченных в управление и руководство проектом. Циклы групп процессов в Руководстве PMBOK® при использовании в проекте с несколькими фазами, этапами в PRINCE2®, ежедневными, еженедельными и ежемесячными циклами в P3.express, итерациями и временными рамками в DSDM® и спринтами в Scrum являются примерами этой концепции.

Более короткие циклы легче понять и использовать, чем более длинные; например, спринты в Scrum в сравнении с фазами из руководства PMBOK®. Однако слишком короткие циклы подходят не для всех типов проектов, решением может быть использование сочетания нескольких типов циклов, например

короткие таймбоксы в сочетании с более длинными итерациями в DSDM®, или использование ежедневных, еженедельных и ежемесячных циклов в P3.express.

Пример: методы

Использование методологии или фреймворка для запуска проекта - еще один пример использования воспроизводимых элементов. Это может быть уже существующая система, такая как PRINCE2®, P3.express, DSDM® или Scrum, или система, которую вы настроили или создали самостоятельно. Однако, как правило, лучше начать с одного из существующих методов и адаптировать его к своим потребностям, чем создавать свой с нуля.

Любой повторяемый элемент является абстрактным и нуждается в настройке, чтобы адаптировать его к реальному миру. Тем не менее, существуют разные степени абстракции и потребности в

адаптации: небольшие, относительно конкретные чек-листы качества находятся на одном конце спектра с наименьшим уровнем абстракции и потребностью в адаптации, в то время как методологии находятся на другом конце, с самой высокой потребностью в адаптации. Вы должны всегда учитывать необходимость адаптации, в противном случае воспроизводимый элемент не будет соответствовать вашим потребностям.