



Требования

# Особенности разработки требований к программным системам

- Разработка требований – это первый из основных процессов создания программных систем. Этот процесс состоит из следующих основных этапов



# Общие положения

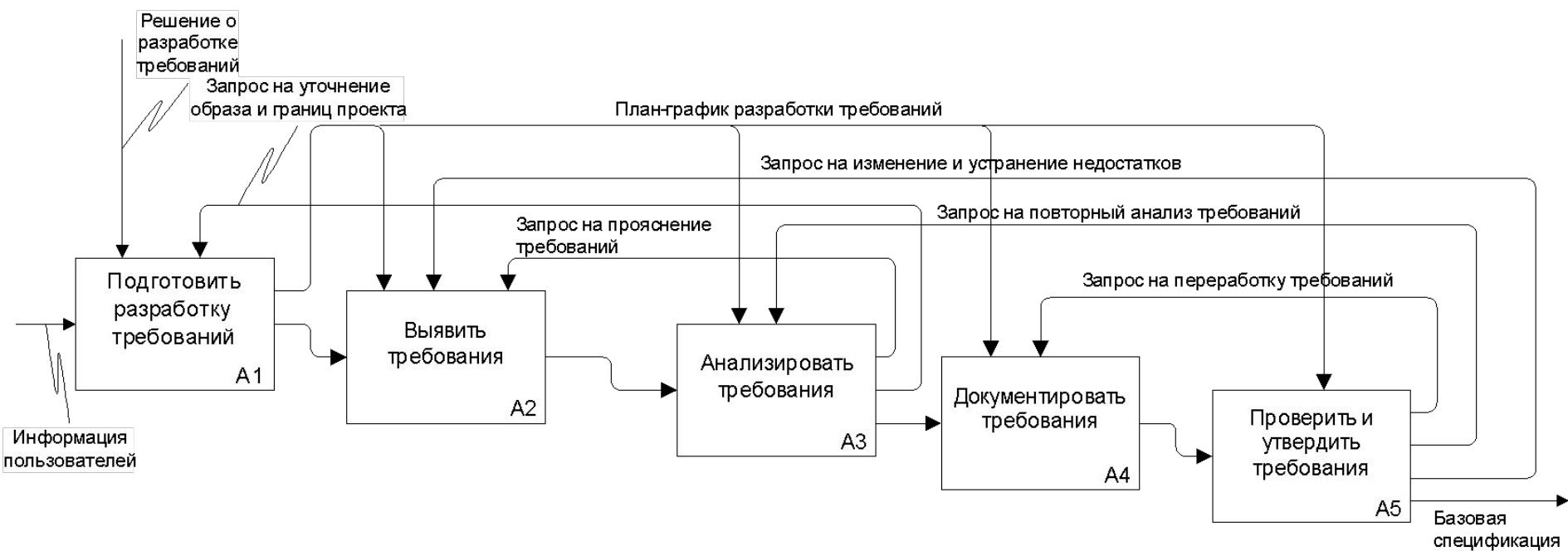
---

Разработку требований можно подразделить на подготовку, выявление, анализ, документирование и проверку. В эти процедуры по существу входят все действия, включающие сбор, оценку и документирование требований для программного обеспечения, в том числе:

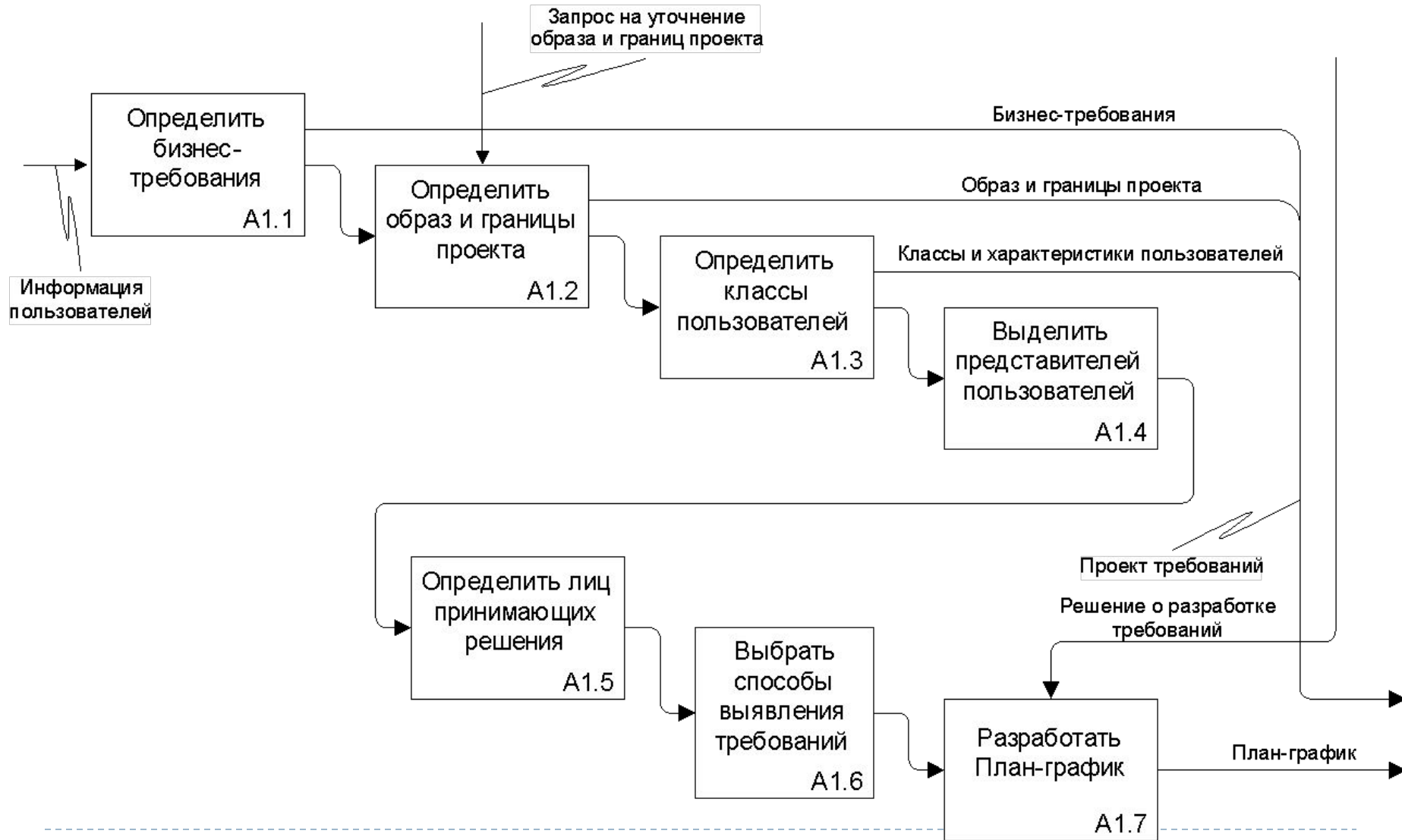
- определение бизнес-требований
  - определение образа и границ проекта
  - идентификация классов пользователей продукта
  - выяснение потребностей тех, кто представляет каждый класс пользователей
  - определение задач и целей пользователей
  - анализ информации, полученной от пользователей, чтобы отделить задачи от функциональных и нефункциональных требований, бизнес-правил, предполагаемых решений и поступающих извне данных
  - установление относительной важности атрибутов качества
  - установление приоритетов реализации
  - документирование собранной информации и построение моделей
  - просмотр спецификации требований, который позволяет удостовериться в том, что запросы пользователей всеми понимаются одинаково
  - устранение всех возникших проблем до передачи документа разработчикам
- 



# Общая модель разработки требований



# Подготовка разработки требований



# Определение бизнес-требований

---

- Бизнес-требования определяют основные задачи и цели проекта, в рамках которых определяются все детальные требования к проекту и являются основой раздела об образе и границах проекта спецификации требований являются

## Подходы к разработке

- бизнес-требования разрабатываются заказчиком самостоятельно:
  - Недостатки: зачастую бизнес-требования, разработанные заказчиком, не обладают требуемыми характеристиками, что в последствии, при разработке детализированных требований, приводит к конфликтам и необоснованным затратам времени и ресурсов
  - Исключительная прерогатива изменения бизнес-требований принадлежит заказчику.
- **совместная разработка бизнес-требований на начальной стадии процесса разработки требований:**
  - информация, касающаяся бизнес-требований, должны поступать от лиц, четко понимающих, почему они взялись за проект. Это может быть непосредственный заказчик и/или его топ-менеджер.

## При разработке бизнес-требований необходимо:

- формулировать бизнес-цели в количественном и измеряемом виде, иначе по завершении проекта будет очень сложно доказать его успешность
  - формулировать критерии успеха, в соответствии с которыми заинтересованные лица будут определять и измерять успех проекта
  - установить факторы, которые максимально влияют на успех проекта – и те, которые вы можете контролировать, и те, которые находятся вне сферы вашего влияния
  - четко определить меру для оценки того, были ли достигнуты бизнес-цели
- 



# Определение образа и границ проекта

---

## Образ продукта

- призван выстраивать работу всех заинтересованных лиц в одном направлении
- должен четко определять, что представляет собой продукт как в базовой, так и в последующих версиях
- что говоря об образе, мы подразумеваем весь продукт
- изменяется относительно медленно при изменении со временем стратегии продукта или развитии бизнес-целей
- сбалансированный образ продукта, удовлетворяющий различные заинтересованные лица, должен обобщать долгосрочные цели и назначение нового продукта. Он может быть несколько идеалистичным, но должен быть основан на существующих или предполагаемых рыночных факторах, архитектуре предприятия заказчика, стратегическом направлении развития предприятия заказчика и ограничениях ресурсов

## Границы проекта

- показывают, к какой области конечного долгосрочного образа продукта будет направлен текущий проект
- относятся к определенному проекту – базовой или последующей версиям. То есть границы более динамичны, чем образ, так как менеджер проекта изменяет содержимое каждой версии в соответствии с графиком, бюджетом, ресурсами и ограничениями качества
- в положении о границах должна быть четко определена черта между тем, что входит в проект и тем, что остается вовне. То есть указанные рамки по существу определяют ограничения проекта

# Ключевые слова для определения образа продукта

Ключевое слово	Образ
<b>для</b>	целевая аудитория покупателей продукта
<b>который</b>	потребности, удовлетворяемые продуктом, и/или возможности, предоставляемые продуктом
<b>эта (этот)</b>	наименование продукта
<b>является</b>	категория продукта
<b>который(ая)</b>	ключевое преимущество, основная причина для покупки или использования
<b>в отличие от</b>	основной конкурирующий продукт, текущая система или текущий бизнес-процесс
<b>наш продукт</b>	положение об основном отличии и преимуществе нового продукта





# Определение образа и границ проекта

- Как уже отмечалось, границы проекта определяют концепцию и круг действия предложенного решения. В ограничениях указываются определенные возможности, которые не будут включены в продукт. Рамки и ограничения помогают установить реалистичные ожидания заинтересованных лиц.
- Определение границы между тем, что входит и выходит за границы проекта, – отличный способ управления расползанием объема и ожиданиями клиентов.
- На практике обычно определение границ конкретной версии осуществляется перечислением бизнес-требований, которые предполагается в рамках этой версии реализовать.
- При поэтапной разработке базовая версия не обязательно должна быть быстрой, красиво оформленной или легкой в использовании, но она должна быть надежной; это основа работы команды. Необходимо помнить, что первая версия системы выполняет лишь базовые задачи. В будущие выпуски будут включены дополнительные функции, возможности и средства, обеспечивающие легкость и простоту использования.
- Для более наглядного представления границ проекта желательно использовать контекстную диаграмму. Она определяет оконечные элементы, расположенные вне системы, которые определенным образом взаимодействуют с ней, а также данные, элементы управления и материальные потоки, протекающие между оконечными элементами и системой. Контекстная диаграмма, по сути, представляет собой высший уровень абстракции в диаграммы потока данных. На подобной диаграмме нет смысла показывать внутренние объекты системы, процессы и данные – она вполне может быть представлена в виде единого объекта. Оконечные элементы обычно представляют классы пользователей, подразделения, другие системы или аппаратные комплексы. Главное назначение подобной диаграммы – наглядно показать, что находится за границами проекта.



# Определение классов пользователей и их характеристик

---

Чтобы не упустить из виду потребности отдельных пользователей, их необходимо подразделить на характерные группы – классы пользователей.

Пользователей продукта можно подразделять по таким признакам:

- по частоте использования продукта
- по опыту в предметной области и опыту работы с компьютерными системами
- по требуемой им функциональности
- по задачам, которые им приходится выполнять
- по правам доступа к системе (например, обычный пользователь, гость или администратор)

На основе подобных признаков и формируются классы пользователей.


При формировании классов пользователей также необходимо учитывать, что некоторых пользователей можно отнести к нескольким классам. Так, администратор иногда работает с системой, как рядовой пользователь.

---



# Определение классов пользователей и их характеристик

---

- ❑ Очень заманчиво поделить пользователей на классы по их географическому положению, виду бизнеса, которым занимается компания, или занимаемой должности, а не на основании того, как они взаимодействуют с системой. В действительности же таким образом выделяются потенциальные сегменты рынка, а не различные классы пользователей.
  - ❑ Одни классы пользователей для вас важнее других. Когда вы принимаете решения о приоритетах или пытаетесь найти компромисс требований, выдвигаемых различными классами пользователей, мнение привилегированных классов имеет первостепенное значение. К последним относятся группы пользователей, работа которых с продуктом определяет, способствует ли он достижению заявленных бизнес-целей или нет. Это отнюдь не означает, что заинтересованных лиц, оплачивающих разработку системы (они, вполне вероятно, вообще не являются ее пользователями), или тех, кто имеет большое политическое влияние, следует обязательно включать в привилегированные классы.
  - ❑ Непривилегированные классы составляют те пользователи, которые по причинам безопасности, конфиденциальности или правовым причинам используют ограниченную функциональность продукта. Остальные классы пользователей можно проигнорировать. Они получают то, что получится, то есть при разработке системы вам не надо учитывать их интересы.
  - ❑ Это может показаться необычным, однако классы пользователей не обязательно состоят из людей. В качестве дополнительных классов пользователей можно рассматривать сторонние приложения и аппаратные компоненты, с которыми взаимодействует ваша система.
  - ❑ Классов пользователей не должно быть слишком много, но в то же время чрезвычайно опасен пропуск какого-либо реально существующего класса.
  - ❑ Определяя классы пользователей, необходимо четко указать их отличительные черты, меру ответственности и физическое расположение.
- 

# Определение представителей пользователей

---

- В процессе выявления требований аналитикам придется в той или иной мере общаться с большинством пользователей, но для эффективной разработки требований необходимо более тесное взаимодействие. Естественно организовать такое взаимодействие со всеми пользователями вряд ли возможно. Более того – в этом случае вы неизбежно столкнетесь с проблемами организации работ в больших группах. Поэтому общепринятой практикой является выделение в каждом классе представителей для участия в разработке требований.
  - По существу – это человек, который сможет точно передавать настроения и нужды соответствующего класса пользователей и принимать решения от их лица. Естественно, такой выбор можно сделать только после определения классов пользователей. Выбранные вами люди должны принимать постоянное участие в проекте (они включаются в состав Группы разработки требований) и обладать полномочиями для принятия решений, касающихся пользовательских требований.
- 



# Определение лиц, принимающих решения

---

- Вполне понятно, что кто-то должен устранять конфликты между требованиями разных классов пользователей, сглаживать противоречия и решать возникающие вопросы относительно границ проекта. Поэтому уже на ранних стадиях проекта необходимо определить тех, кто будет принимать решения, касающиеся требований. Если не ясно, кто за это отвечает, или уполномоченные лица отказываются брать на себя такую ответственность, в конечном итоге принятие решения возлагается на разработчиков. Это – неудачный выход, поскольку у разработчиков обычно нет необходимых знаний, опыта и стратегического видения для принятия оптимальных бизнес-решений.
  - Не существует единственно верного ответа на вопрос о том, кто должен принимать решения по требованиям к программному проекту. Однако, по возможности, решения должны принимать сотрудники низких уровней организационной иерархии, которые непосредственно занимаются выпуском продукта, и пользователи, которые непосредственно будут работать с продуктом.
  - Для упрощения процедуры принятия решений также необходимо определить соответствующие правила решения.
- 



# Выбор способов выявления требований

---

- Существует большое множество апробированных способов выявления требований. Однако далеко не все из них универсальны и применимы в любых ситуациях. Одни способы ориентированы на ситуации, когда разрабатываемый проект является частью другого высокоуровневого проекта, другие – на ситуацию, когда разрабатываемый продукт представляет собой некоторую разновидность уже существующих продуктов, третьи – на ситуацию, когда проект направлен на модернизацию существующего продукта и т.п. Иными словами, выбор способов выявления требований существенно зависит от специфики конкретного проекта.
  - С другой стороны способы выявления требований различаются по затратам, требующимся на их осуществление. Следовательно, нельзя указать какой-либо набор способов выявления требований, применимый в любых ситуациях, в особенности, учитывая существующие ограничения по использованию ресурсов.
  - Поэтому весьма важным является правильный выбор способов выявления требований на начальной стадии разработки требований с учетом специфики разрабатываемого проекта. Конкретный набор способов выявления требований фиксируется в Плане-графике разработки требований и служит основанием для планирования необходимых мероприятий и выделения необходимых ресурсов.
- 



# Разработка Плана-графика

---

План-график разработки требований является основным документом, регламентирующим разработку требований. В отличие от Решения о разработке требований, устанавливающего лишь основные цели и задачи разработки требований и фиксирующего создание Группы разработки требований, План-график разработки требований:

- устанавливает конкретный состав Группы разработки требований и распределение обязанностей среди ее участников;
  - устанавливает перечень лиц, принимающих решения относительно требований;
  - определяет правила принятия решений;
  - определяет перечень используемых способов выявления требований;
  - определяет перечень работ и мероприятий по разработке требований, сроки их проведения и участников.
- 



# Разработка Плана-графика

---

- План-график разработки требований также является основанием для выделения необходимых ресурсов.
- В зависимости от сложности и масштаба проекта, а также используемых технологических средств планирование работ может осуществляться в виде простого календарного плана, временных диаграмм, диаграмм Гантта или PERT-диаграмм.
- При планировании работ необходимо учесть ряд важных моментов, не только часто не учитываемых, но и зачастую не осознаваемых.





# Разработка Плана-графика

---

- Планы, как правило, расходятся с действительностью. Поэтому современные методологии так и говорят - да, планы будут меняться, да, этого не избежать, просто нужно организовывать процесс так, чтобы минимизировать потери и минимально снизить стоимость внесения изменений в проект. Именно этой проблеме обязаны своим появлением целое семейство "agile" методологий (XP - яркий тому пример), Microsoft-овский MSF раскололся на две ветви, Agile и Formal, да что там, даже у тяжеловесного RUP появились agile направления, например dX, разработанный одним из авторов Agile-манифеста Робертом Мартином.
- Методологии существуют, но вот цитата из доклада "Хаос" компании Standish Group (статистика по 364 американским компаниям и 23 тысячам проектов): "Только 16.2% проектов завершились в срок, не превысили запланированный бюджет и реализовали все требуемые функции и возможности; 52.7% проектов завершились с опозданием, расходы превысили запланированный бюджет, требуемые функции не были реализованы в полном объеме; 31.1% проектов были аннулированы до завершения. Для проектов, которые завершились с опозданием или были аннулированы до завершения, бюджет среднего проекта оказался превышенным на 89%, а срок выполнения - на 122%".



# Допущение первое: в 8-ми часовом рабочем дне 8 рабочих часов

---

- При планировании аксиоматично под рабочими часами понимается то время, когда человек занят именно работой: кодирует, пишет документацию, принимает участие в совещаниях и т.п. Но все мы понимаем, что указанные выше 8 часов далеко не "рабочие".
  - Так сколько же рабочих часов в 8-ми часовом рабочем дне? Ведь для того, чтобы найти ответ, не достаточно просто использовать какую-нибудь систему учета времени. Как правило, рассчитанные по ней результаты идеально совпадают с 8 часами. Странно? Вовсе нет. Большинство систем учета времени построено на самооценках, причем зачастую некая форма заполняется **после** выполнения работ (в конце рабочего дня или на следующий день). Сотрудник абсолютно искренне пытается **разделить** отработанные 8 часов на задачи, которые он решил. И даже если фиксировать время чаще, точнее становится только список задач, идея обычно остается прежней - прикинуть, как разделить между ними отработанные 8-мь часов.
  - Причем следует заметить - это не попытка скрыть время, когда человек бездельничал. Вовсе нет! Это просто психологический парадокс. Недаром в более приземленных, не столь "высокотехнологичных" отраслях за станочником обычно наблюдал нормировщик, а не он сам заполнял форму!
  - Люди не роботы и избежать потерь времени **невозможно!** Этого не удастся сделать даже на конвейере при выполнении рутинных, механических операций. Что же можно сказать о работе с высоким уровнем "творческого" компонента. Да, мы отвлекаемся, хотя часто это тоже работа. Просто другая. Мы разбираемся с коллегами, с их проблемами, советуем что-то, иногда возникают срочные задачи, и мы переключаемся на них. И это тоже правильно – вы объяснили что-то соседу и сэкономили ему пару часов времени. А завтра он посоветует что-то вам. Вы отвлеклись на возникшую проблему, но иначе простаивало бы несколько человек...
  - По статистике, в наиболее продвинутых компаниях с жестко поставленным процессом и направленной на это корпоративной культурой за день работе отдается до 7-и часов. Это лучший результат. Том Демарко в своей культовой книге "Peopleware: Productive Projects and Teams" приводит цифру 75%, т.е. примерно 6 часов, как пример хорошей организации. Хотя существует достаточно много организаций, где на работу тратится 5, а то и 4 часа в день.
- Вывод первый:** составляя планы, не забывайте, что даже при хорошем раскладе в рабочем дне не восемь, а скорее всего лишь 6 часов, на которые вы и можете рассчитывать.
- 



## Допущение второе: все рабочее время будет затрачено на выполнение запланированных работ

---

- Один из законов Мерфи гласит: "Планируя что-нибудь, отведите время на непредвиденные ситуации, а затем - еще больше времени на непредвиденность самих непредвиденных ситуаций".
- Заметим, речь идет не о рисках - это отдельный вопрос. Непредвиденные ситуации - это поломка локальной сети, пропавшее на день электричество или замолчавшие телефоны, сбой или поломка компьютера, администратор, забывший настроить резервирование базы данных и прочие мелкие "радости" жизни. Не следует также забывать, что люди, занятые в проекте, должны отвечать еще и за другие дела (с началом проекта жизнь в организации-разработчике не останавливается).
- Обычно запас времени на непредвиденные ситуации вычисляют в процентах от общей продолжительности проекта. Естественно конкретная величина зависит от множества факторов, но более или менее разумной величиной принято считать 5-6%.

**Вывод второй:** составляя планы, не забывайте включать в них запас по времени для непредвиденных ситуаций.

---



# Еще одна проблема утечки времени: хаотичность поддержки

---

Эта проблема специфична именно для организаций - разработчиков программного обеспечения. Если только планируемый проект у вас не первый, у вас неизбежно существует необходимость поддержки реализованных проектов.

Время от времени у каждого из ваших заказчиков возникают вопросы - он чего-то не знает, сделал что-то неправильно и теперь не знает, как это исправить, он думает, что у него проблемы с приложением, хотя на самом деле - это проблемы с аппаратурой... или все-таки с приложением? Да мало ли. Подобные вопросы возникают не так уж часто, поэтому тратить чье-то время на постоянную поддержку просто жалко. Да и работа это не самая интересная и добровольцы находятся редко. Более того, обычно заказчик звонит или пишет человеку, с которым он привык общаться, - например, менеджеру проекта или аналитику, чем несказанно их радует. Причем заказчик всегда считает, что его проблемы критичны по времени и легко приходит в раздражение, услышав слова "обратитесь в пятницу" или что-то вроде этого...

По мере роста компании рано или поздно все же приходят к необходимости создания соответствующего подразделения поддержки, остальные же зачастую пускают решение вопросов поддержки на самотек, что и порождает незапланированные "утечки" времени.

Как с этим можно бороться? Вот несколько организационных приемов, могущих если не минимизировать потери времени на поддержку, то хотя бы сделать их прогнозируемыми (а, следовательно, такими, которые можно учесть):

- Все-таки выделить сотрудника (сотрудников), который будет тратить часть своего времени на поддержку реализованных проектов.
- Ввести график обращения по вопросам поддержки и ознакомить с ним своих заказчиков
- Разработать шаблон документа, который необходимо заполнить при обращении по вопросам поддержки (Это поразительно! Если вместо простого набора номера и разговора "по душам" необходимо затратить всего 10-15 минут на заполнение формы, количество вопросов существенно сокращается!).

**Вывод третий:** не оставляйте вопросы поддержки на самотек. Или создайте службу поддержки и избавьте сотрудников, занятых в проекте от решения вопросов поддержки, или организуйте поддержку так, что бы затраты времени на нее можно было учесть.

---

