

**Конспект лекций
по дисциплине: «Управление проектами...»**

Тема 1. Жизненный цикл ИС. Место процесса проектирования в жизненном цикле.

Схема этапов жизненного цикла. Факторы, влияющие на концепции эволюции жизненных циклов. Место «проектирования» в жизненном цикле ИС. Систематизация задач поддержки процесса разработки ИС. Системная инженерия - процессы жизненного цикла систем (ISO/IEC 15288).

Схема этапов жизненного цикла. Отметим следующее определение: «Современная индустриальная технология проектирования программ включает в себя комплекс мероприятий, руководящих документов и автоматизированных средств, предназначенных для системного анализа, разработки, отладки, документирования, управления работой специалистов и контроля эксплуатации программ».

Немаловажную роль в определении процесса инженерного проектирования играет структура жизненного цикла программных систем (см. рис.2.1), которая позволяет очертить место этого процесса в общей картине эволюции разрабатываемой системы, а также учитывать ряд влияющих внешних факторов, в том числе:

- экономические – оценка рынка (спроса и предложения), учет производственных затрат и т.д.;
- эргономические – учет эстетичного вида и медицинской безопасности интерфейса не только самой программной системы, но и учет товарной привлекательности готового продукта;
- психологические – учет психологии заказчика или покупателя, а также человеческого фактора в лице специалиста(-ов) по предметной области в процессе постановки задачи.

В научной литературе, как правило, процесс разработки ПС носит структурированный характер, выражающийся в разбиении его на стадии и этапы. При этом, большинство подходов к такому разбиению ограничивается манипулированием и последовательной фиксацией названий стадий как элементов из конечного множества терминов: техническое задание, эскизный проект, технический проект, рабочий проект, внедрение и дальнейшей их детализации через описание этапов. Разбиение на этапы также характеризуется индивидуальностью авторских подходов. Можно только выделить наиболее повторяющиеся названия этапов проектирования: системный анализ и проектирование алгоритмов, структурное проектирование, подготовка технологических средств, разработка программ, отладка программ в статике, комплексная динамическая отладка, выпуск машинных носителей и документирование, испытание программных средств и прочее. При этом за основу каждого этапа берутся разные классы моделей представления и, соответственно, разные языки описания этих моделей, и разный формальный аппарат анализа и синтеза (текстовое описание, графические нотации, математические модели и др.).

Как видим, к сожалению, структуризация стадий и этапов процесса проектирования, выбор моделей представления носит субъективный характер. Из исследований материалов по этому вопросу можно отметить только существование зависимости стадий и этапов от сложности разрабатываемой программной системы.

С учетом выше изложенного, напрашивается вывод о необходимости разработки научно-обоснованного обобщения, методики, стандарта выбора структуры стадий и этапов процесса проектирования с увязкой к формальному аппарату модельного представления программных систем.

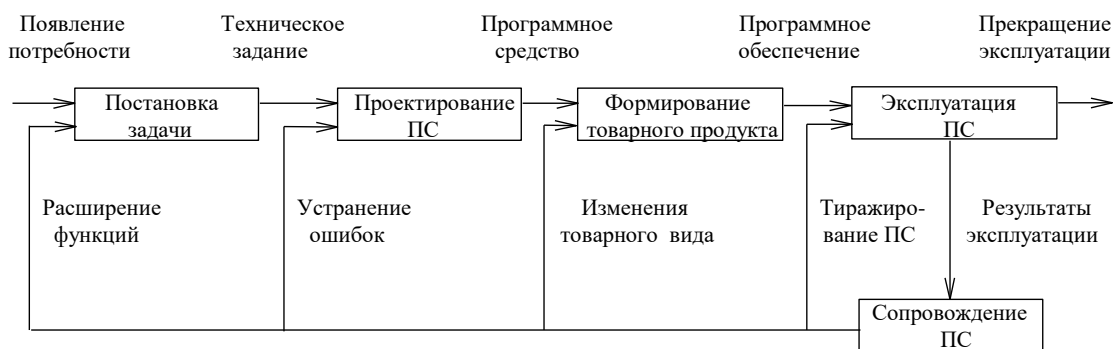


Рис. 2.1. Обобщенная схема этапов жизненного цикла.

Системная инженерия - процессы жизненного цикла систем (ISO/IEC 15288:2008).

Стандарт ISO/IEC 15288 «Системная инженерия - процессы жизненного цикла систем».

В мире наблюдается постоянный рост сложности систем, создаваемых человеком. Такое усложнение приводит к ряду новых проблем, возникающих на всех стадиях жизненного цикла системы и на различных уровнях ее архитектурной детализации. Источниками проблем служат разнородность составных элементов системы (оборудование, люди, ПО), комплексное использование компьютерных технологий, недостаточная интеграция применяемых дисциплин. Для преодоления возникающих проблем требуется общий подход, обеспечивающий эффективное взаимодействие лиц, которые создают, используют и управляют современными системами. Основными группами задействованных лиц являются менеджеры, управляющие созданием систем, и инженеры, создающие системы. Стандарт ISO/IEC 15288 в качестве подхода, объединяющего эти группы, предлагает общий набор практик, охватывающий весь жизненный цикл рукотворных систем, и предписывает при работе со сложной системой иметь описание ее жизненного цикла.

Определение стандарта гласит: **жизненный цикл (ЖЦ)** – это эволюция системы, продукции, услуги, проекта или иного рукотворного объекта от замысла до прекращения использования. Каждая система, вне зависимости от ее вида и масштаба, проходит весь свой жизненный цикл согласно некоторому описанию. Продвижение системы по частям этого описания и есть жизненный цикл системы. Описание жизненного цикла, таким образом, — это концептуальная сегментация по стадиям, способствующим планированию, разворачиванию, эксплуатации и поддержке целевой системы. Стадии представляют наиболее крупные периоды жизненного цикла, ассоциируемые с системой, и соотносятся с состояниями описания системы или реализацией системы как набора продуктов или услуг. Стадии описывают основные контрольные точки продвижения и успехов системы по ходу жизненного цикла. Такие сегменты дают упорядоченное продвижение системы через установленные пересмотры выделения ресурсов, что снижает риски и обеспечивает удовлетворительное продвижение. Основной причиной применения описаний жизненного цикла является потребность в принятии решений по определенным критериям до продвижения системы на следующую стадию.

Жизненный цикл неотделим от конкретной системы, поэтому особенности разных систем порождают большое разнообразие экземпляров жизненных циклов. Управленцы, в зависимости от выбранной стратегии и профиля существующих рисков, применяют различные последовательности стадий, что приводит к формам жизненного цикла с различающимися характеристиками. К ним можно отнести последовательную, инкрементальную, эволюционную формы. Инженеры, в свою очередь, используют различные формы разработки системы – восходящую, нисходящую, изнутри-наружу (middle-out). Были созданы методы управления жизненным циклом, представляющие собой типовые описания форм жизненного цикла в их связи с формами разработки, нацеленные на использование в определенных условиях. Наиболее распространенными методами являются RUP, Agile, DSDM, V-model, ICM.

Базовые характеристики системной инженерии в разрезе стандарта:

- Междисциплинарный подход к проблеме успешного создания систем и средство для её решения.

- Фокусируется (при постоянном внимании к охвату проблемы во всей полноте):
 - на определении нужд пользователей и требуемой функциональности на ранних стадиях цикла разработки,
 - на документировании требований,
 - на синтезе дизайна системы,
 - на подтверждении соблюдения пользовательских требований
- Описывает процесс разработки систем и как бизнес-процесс, и как технический процесс
- Охватывает стадии жизни систем от появления замысла до вывода из эксплуатации

Основное назначение и задачи стандарта:

- Гармонизировать многочисленные стандарты, принятые различными военными ведомствами, государствами, отраслевыми организациями стандартизации.
- Дать возможность организациям (внешним и внутренним контракторам) договориться о совмещении замыслов, процессов проектирования, создания, эксплуатации и вывода из эксплуатации самых разных рукотворных систем – от зубочисток до атомных станций, от систем стандартизации до корпораций
- Внедрить в практику организации ряд ключевых идей системной инженерии:
 - системного подхода
 - жизненного цикла
 - инжиниринга требований
 - архитектурного дизайна
 - процессного подхода
 - проектного подхода
 - культуры контракции

Общие свойства стандарта:

- Применим к любым рукотворным системам любой области человеческой деятельности (включая организации, сервисы, сами системы стандартизации).
- Охватывает полный цикл жизни (например: замысел, разработка, производство, использование, поддержка и вывод из эксплуатации)
- Учитывает необходимость контракции (приобретения и поставки продуктов и услуг)
- Охватывает использование внутри организаций и между организациями (в «расширенной организации» проекта)
- Включает в процессы людей, оборудование, компьютеры, софт (ссылается на связанный стандарт ISO 12207 – жизненный цикл программного обеспечения)
- Применяется **параллельно, итеративно и рекурсивно** для различных частей системы
- Учитывает особенности композиции любых систем – встроенных, автономных, интегрированных и любых других, сложных и простых
- Обладает технологической нейтральностью - не указано, какие именно технологии необходимо использовать в предписанных процессах, внедрение стандарта состоит во внедрении конкретных технологий/методологий получения результатов, требуемых стандартом (в управлении моделью жизненного цикла – выбор из IDEF, UML, DEMO, BusinessStudio, OrgMaster и т.д.)

Жизненный цикл:

- Стадии жизненного цикла (жёстко стандартом не предписываются)
- Стадии могут сосуществовать или возобновляться после завершения
- Стадии определяют основные точки принятия решений, вехи проекта (decision gates)
- Понятие зафиксированного состояния (baseline): после формального закрепления принятых по проекту решений, изменение которых можно провести только через формальную процедуру

Практикуется методология V – модели (рис. 1).

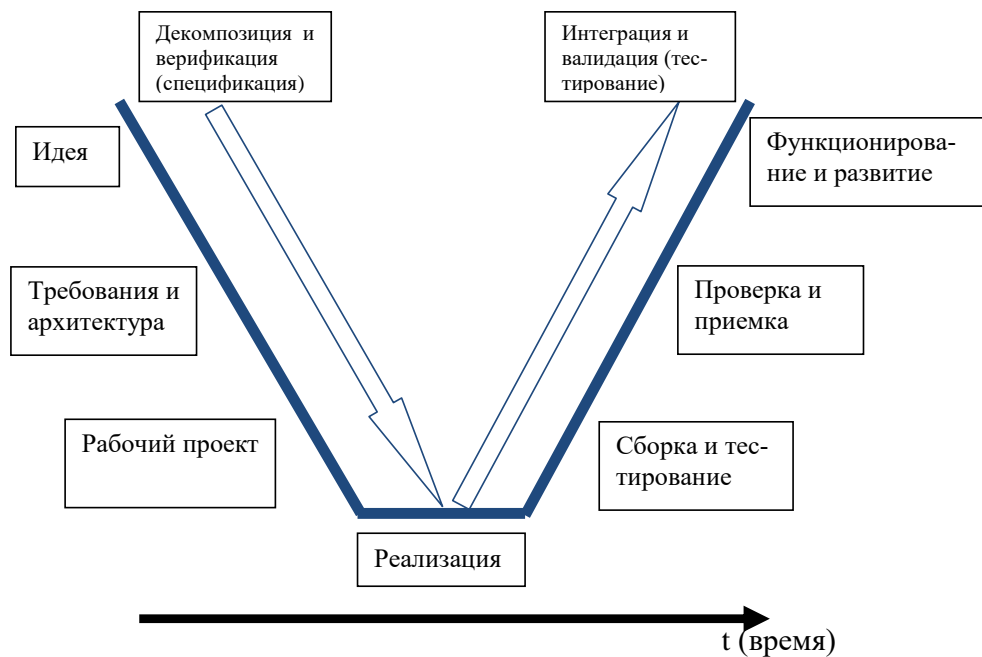


Рис. 1. V-модель жизненного цикла.

Разнообразие жизненных циклов

1. Программное обеспечение: Концепция-Разработка-Поддержка-Списание.
2. Оборудование: Идея-Проектирование-Изготовление-Эксплуатация и поддержка-Списание.
3. Процесс: Определение выхода-Графическое представление-Описание-Пилотное внедрение-Использование и совершенствование-Ликвидация.
4. Система: Идея-Разработка-Изготовление-Использование-Поддержка-Списание.
5. И др.

Инжиниринг требований. На всех стадиях жизненного цикла:

- Определять заинтересованных лиц и собирать их требования (требования стейкхолдеров)
- Анализировать требования стейкхолдеров и формировать требования разработчиков
- Проводить проверку соответствия требованиям разработчиков (верификацию)
- Проводить пользовательскую приёмку (валидацию)

Архитектурный дизайн:

- Архитектура – фундаментальная организация системы, воплощённая в её компонентах, их отношениях друг с другом и с внешним окружением, а также принципы, управляющие её созданием и развитием (ISO/IEC 42010:2007, IEEE 1471)
- Архитектура задаёт решение *проблемы* (необходимость соответствия противоречивым требованиям) в виде описания взаимодействующих подсистем
- Архитектурный *синтез* после *анализа* требований
- Архитектурное описание – набор *взглядов* и *моделей*, отражающих различные формально зафиксированные *точки зрения стейкхолдеров*
- Создание архитектуры – стадия жизненного цикла системы, имеющая свой жизненный цикл (например, замысел архитектуры, определение требований к архитектуре, разработка архитектуры, создание архитектуры, использование архитектуры), и содержащая все процессы ISO 15288

Процессный подход:

- **Процессный подход** – рассмотрение всех действий в организации как реализаций **типовых шаблонов**, отражающих те или иные организационные практики (PraxOS)
- Процесс – набор взаимосвязанных или взаимодействующих действий, преобразующих входы в выходы (из ISO 9000:2005). Процессы состоят из действий (activities), а действия – из задач (tasks)
- Процессы – форма привнесения, фиксации и развития организационных практик
- Процессы документированы, могут быть предметом коммуникации (обсуждения)
- Выполнение процессов проверяемо (ISO 15504):
 - Наличие продуктов (результатов процессов)
 - Выполнение всех предписанных действий и задач
 - Выполнение требований и ограничений
 - Выполнение измерений хода процесса

25 обязательных процессов системной инженерии

Обеспечения проектов

1. управление моделью жизненного цикла
2. управление инфраструктурой
3. управление портфелем проектов (программой)
4. управление персоналом
5. управление качеством

Проектные

управление проектами

6. планирование проекта
7. управление выполнением
8. контроль проекта

поддержка проектов

9. управление решениями
10. управление рисками
11. управление конфигурацией
12. управление информацией
13. измерения

Технические

14. анализ требований
15. архитектурный дизайн
16. изготовление
17. интеграция
18. проверка (Verification)
19. переход к эксплуатации
20. приёмка (Validation)
21. эксплуатация
22. обслуживание
23. вывод из эксплуатации

Контрактацции

24. Закупка
25. Поставка

Для решения поставленных задач в рамках стадий ЖЦ выполняются практики. **Практика ЖЦ** – это набор взаимосвязанных и взаимодействующих мероприятий, перерабатывающий исходные сущности в конечные. Вследствие действий, управляемых и исполняемых людьми в одной или нескольких организациях, применяющих практики, выбранные для данной стадии жизненного цикла, происходит продвижение системы по ее циклу. По сути практики описывают то, как выполнять работу, продвигающую систему по ЖЦ. Существующие стандартные методы включают в себя описание предлагаемых практик.

Каждая из практик жизненного цикла, предложенная тем или иным методом, может при необходимости применяться в любой точке ЖЦ, и в их применении нет предопределенного порядка или последовательности. Конкретное назначение и время применения этих практик на протяжении ЖЦ зависит от множества факторов, включая социальные, коммерческие, организационные и технические соображения, каждое из которых может изменяться на протяжении жизни системы. ЖЦ конкретной системы, таким образом, представляет собой сложную систему практик, обычно обладающих характеристиками параллельности, итеративности, рекурсивности и зависимости от времени.

Стандарт предписывает, что организации, приступая к новому проекту, должны выбрать описание жизненного цикла и практики жизненного цикла, необходимые для удовлетворения входным или выходным критериям применимой стадии жизненного цикла. Решения, касающиеся выбора практик, должны основываться на анализе выгод и затрат или минимизации рисков.

Описать жизненный цикл, значит описать его **экземпляр**, представленный связанным набором элементов жизненного цикла. Экземпляр жизненного цикла определяется методом управления жизненным циклом путем конкретизации элементов. Элементы включают в себя практики, контрольные точки, пересмотры выделения ресурсов, артефакты, роли, инструменты, руководства. Описание документирует решения принятые в процессе конкретизации.

Выделяются несколько разных групп описаний, ориентирующихся на различные элементы. Прежде всего – это организационная и инженерная группы.

Организационная группа описаний включает в себя **рассмотрение типовых контрольных точек** и пересмотров выделения ресурсов, а также формы жизненного цикла. Такое описание позволяет удовлетворить потребность в принятии решений, связанных с переходом между стадиями ЖЦ. Инженерная группа описывает применение практик в рамках стадий ЖЦ, а также выбранную форму разработки системы, степень итеративности разработки, технические пересмотры, объем валидации архитектурных решений и другие аспекты. Эта группа описаний удовлетворяет потребность инженеров в описании методов работы.

По сути, описание ЖЦ – это описание деятельности в виде наборов актов деятельности (микропроцессов), т.е. практик, синхронизированных и выполняемых в рамках последовательности стадий (макропроцессов). Несмотря на то, что стандарт предписывает иметь описание жизненного цикла, в нем не определяется формат подобного описания.

Когда мы говорим о ЖЦ, то понимаем типовые активности (работы), внутри которых используются те или иные методы, описанные отдельно от их применений в жизненном цикле. Эти описанные отдельно методы, которые в терминологии ISO 15288 названы практиками (в оригинале – process, унаследованное из трансформационного описания), применяются в стадиях ЖЦ, сгруппированных в определенную последовательность и разделенных контрольными точками. Для того чтобы создать полноценное описание ЖЦ и его практик необходимо:

- отразить технологию осуществления проекта, т.е. показать:
 - методы, которые должны применяться;
 - последовательность применения дел, из которых состоят методы;
 - артефакты, связанные с методами;
 - исходные и конечные сущности и информация;
 - роли, ответственные за дела;
 - инструменты, применяемые ролями;

- квалификацию, требуемую для выполнения дел;
 - руководства, используемые при работе;
- отразить сам процесс ЖЦ (его форму) и его результаты, т.е. определить:
- стадии проекта;
 - последовательность стадий во времени;
 - повторяемость стадий (применение итераций);
 - продукты, производимые на каждой стадии;
 - контрольные точки, в которых принимаются решения по проекту.

На основе этого списка видно, как должны поменяться процессная и проектная группы описаний.

В процессной группе происходит смена объекта рассмотрения. Очевиден переход к рассмотрению не только последовательности дел, но и их связанных групп – методов (практик, дисциплин), которые объединены общностью применяемой в их онтологией и зачастую общностью ролей, которые способны эти практики выполнять в силу знания этой онтологии. Методы (практики, дисциплины) также обеспечиваются руководствами по их использованию.

Тема 2. Эффективность проектирования.

Структура задач оценки эффективности ИС. Показатели эффективности. Теория праксеологии в проектировании ИС.

1. Оценка качества инженерной деятельности при разработке программных систем.

При разработке ИС приходится решать проблемы, тесно связанные с оценкой эффективности (например, эффективность производительности фирмы проектировщика или эффективность тиражирования ИС или эффективность трудозатрат и т.д.). Эти проблемы, в свою очередь, тесным образом связаны с комплексом задач (см. рис.1) решение которых зависит от разных групп оценщиков.

– Разработчики ИС – руководители подразделений, системные аналитики, программисты сталкиваются с проблемами, которые можно существенно облегчить, если использовать научно обоснованные технические приемы и оценки их проектной деятельности.

– Потребители ИС – заказчики, покупатели, пользователи, сталкиваются с задачами потребительской оценки приобретаемой ИС.

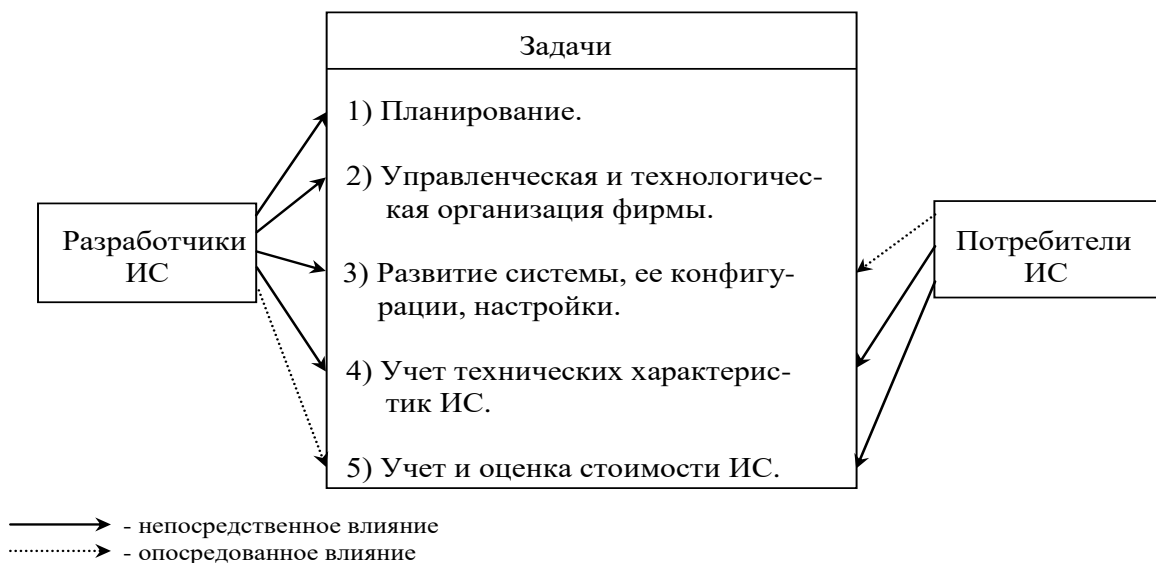


Рис. 1. Структура задач оценки эффективности разработки ИС.

Теория проектирования ИС еще недостаточно разработана, чтобы на любой стадии разработки можно было бы воспользоваться количественными закономерностями для оценки эффективности инженерных решений. Эта ситуация хорошо иллюстрируется почти исключительно описательным, качественным подходом, используемым в современной научной литературе. Такой подход являет собой форму субъективного описания. Научный же подход требует интерпретации субъективных определений в чисто технические термины, которым может быть придана количественная форма и которые, следовательно, можно оценить объективно. Однако ряду важных понятий на сегодняшний день не придумана количественная мера представления (например, структурность представления, легкость использования и др.). Как правило, такие понятия соотносят с функциональными аспектами моделирования ИС.

2. Праксеологический подход к оценке эффективности инженерных решений.

Разработка и совершенствование средств и методов описания ИС направлены, в первую очередь, на повышение эффективности применения средств вычислительной техники в области управления. Существующие методики оценки экономического эффекта в денежном выражении для программной составляющей автоматизированных систем в области управления крайне несовершенны из-за необходимости учета множества недостоверных факторов, поэтому приходится широко использовать качественные оценки.

Введем понятие критериев эффективности информационной технологии (ИТ), как показателей полезности результатов функционирования системы. Любой такой показатель может быть неоднозначным и значительно меняться от условий эксплуатации всей системы в целом.

Подходы к оценке эффективности ИТ будем осуществлять с позиций, что эффективность применения средств вычислительной техники и разработки и совершенствования обеспечивающих средств должна оцениваться по основным критериям функционирования ИС: оперативности, точности и полноты информации, которая предоставляется пользователям для принятия решений. Другую составляющую эффективности ИТ свяжем с совершенствованием информационной работы в фирмах, с организацией документооборота и его ведением, т.о. центр тяжести в оценке общего экономического эффекта сместим в сторону совершенствования системы управления.

С целью дальнейшей оценки эффективности инженерных решений разработки ИС воспользуемся идеями общей теории эффективности действия - праксеологией (анализом потребительских эффектов). Согласно ей эффективность того или иного действия соответствует степени достижения цели при заданных затратах.

Трансформируем идеи праксеологии с целью дистанцирования от методов экономической (рублевой) оценки инженерных решений, а также формирования единого общего подхода к методике как оценки системы технологических знаний (прагматический подход), так и оценки выделенных программных систем (апрагматический подход).

Если термин «эффективность» понимать в универсальном смысле, как общее название любого из практических достоинств, то такие понятия, как точность, производительность и т.д., представляют собой разновидности эффективности. С точки зрения праксеологии выделим три основных вида эффективности.

- Результативность – степень реализации цели (конечной или промежуточной).
- Полезность – разница между ценностью достигнутого результата и затратами на его достижение.
- Экономичность – отношение полезного результата действия к средствам, затраченным на его реализацию.

Праксеология различает и другие виды эффективности: производительность, простоту, эстетичность, точность, рациональность и пр. Однако эти виды эффективности представляют собой больше некие условия (факторы), от которых зависят три выделенных вида эффективности.

Для получения обобщенной полной оценки, необходимо использовать понятие суммарной эффективности. Таким праксеологическим понятием является эффективность в синтезированном виде. Эффективность в синтезированном смысле слова – это совокупность практических достоинств.

Выделим базовые праксеологические параметры.

– Цель действия (сокращенно – цель, обозначение - C) – состояние объекта, которое является ценным (желательным), оцениваемое с некоторой позиции. Цель действия определяет направление и структуру действия субъекта, стремящегося к тому, чтобы создать или сохранить желаемую ситуацию.

– Результат действия (обозначение R) – состояние объекта, достигнутое за счет действия и являющееся результатом реализации цели или ее части.

– Затраты (обозначение – N_C и N_R) – объем ресурсов, планируемых (N_C) и израсходованных (N_R) на реализацию действия для достижения полезного результата.

Цели, результаты и затраты объединяются общим понятием – эффект действия. В рамках этого понятия имеем: цель – предвидимый максимально возможный эффект, результат – положительно оцениваемый достигнутый эффект, затраты – отрицательно оцениваемый эффект. Это дает основание считать, что *размерность вышеуказанных величин одинаковая*.

Однако, определение эффективности в синтезированном смысле наталкивается на ряд трудностей, связанных с необходимостью определения соответствующей меры для каждого из достоинств хорошей работы. Эта проблема сводится к возможности определения неких эталонов (единиц измерения) в замену общепринятого денежного эквивалента и, которыми можно определять эффективность в синтезированном виде без привязки к экономической эффективности. В случае определения такого эталона праксеологические параметры универсальной эффективности примут теоретико-множественный вид (ф 1.1).

$$C = \bigcup_{i=1}^c \xi_i^C, R = \bigcup_{i=1}^r \xi_i^R, N = \bigcup_{i=1}^n \xi_i^N, C \cap R \cap N = \emptyset, \quad (1.1)$$

где ξ - эталон измерения

При таком представлении праксеологических параметров можно определить различные праксеологические показатели (таблица 1.1).

Набор праксеологических показателей не заканчивается на перечисленных в таблице 1.1. В любом случае, в явном или нормированном варианте, праксеологические показатели могут служить основой для разработки показателей качества EDAT.

Таблица 2. Набор праксеологических показателей.

Праксеологический показатель	Наименование	Комментарий
$\sigma = \frac{r}{c} = \frac{ R }{ C }$	Качество результата	Отношение количества эталонов результата действия к количеству эталонов цели действия.
$\delta = r - c = R - C $	Рассогласование результата	Разность количества эталонов результата действия и количества эталонов цели действия.
$\chi = r - n = R - N_R $	Полезность результата	Разность количества эталонов результата действия и количества эталонов затрат на реализацию действия.
$\eta = \frac{r}{n} = \frac{ R }{ N_R }$	Экономичность результата	Отношение количества эталонов результата действия к количеству эталонов затрат на реализацию действия.
$\varepsilon = N_C - N_R $	Рассогласование затрат	Разность количества эталонов планируемых затрат и количества эталонов израсходованных затрат на реализацию действия

Эффективность в синтезированном виде представляет собой некую многопараметрическую функцию:

$$\Xi = \Xi(k_1, \dots, k_n) = \Xi(K), \text{ где } k_i \in K - \text{ праксеологические показатели}$$

Задачу праксиометрической оценки логического моделирования ИТ будем сводить к определению эталона (-ов) измерения и способов измерения праксеологических параметров **R, C, N**.

Изменение критериев качества будем базировать на введении числовых параметров и мер, характеризующих функциональное назначение и конструктивные особенности конкретных ИС.

Выбор экономических и функциональных критериев для описания эффективности создания и использования ИС зависит от их назначения, области применения и других факторов. Преобладает экономический эффект, описываемый суммарным доходом от использования программного средства в течение его жизненного цикла. Этот доход представим как разность между полной идеальной эффективностью применения ИС и суммарными потерями и затратами, снижающими предельный доход. Тогда экономический эффект сопоставим с праксеологическим показателем полезности результата χ .

Результат действия **R** будем рассматривать как совокупный доход от использования ИС, который можно было бы получить при отсутствии затрат на создание (моделирование и стремление получить на этом этапе высокие эксплуатационные характеристики, обеспечить удобное сопровождение и т.д.) и эксплуатацию (последствия сбоев, отказов ПК, ограничения ресурсов по памяти и производительности, неполная отладка программ и т.д.).

Затраты на реализацию действия **N** будем рассматривать как основные затраты, которые представим следующими составляющими:

- совокупные затраты на создание ИС и обеспечения решения заданных функциональных задач, в том числе на технологическое обеспечение и аппаратную оснащенность разработки;
- затраты на эксплуатацию программных средств, реализующих ИС, а также совокупные потери эффективности из-за ограниченных характеристик реализующей вычислительной техники и неидеальных программ;
- затраты на сопровождение ИС, включая затраты на хранение и контроль его состояния, проведение модернизаций, исправление ошибок.

Очертим круг проблем, за счет решения которых в ИТ повышается эффективность процесса разработки ИС:

- развития методической поддержки, включающей в себя комплекс стандартов, инструкций и методик, определяющих правила моделирования, проведения экспериментов, создания программ, документов, регламентирующих построение объекта разработки и процесса его создания;
- усовершенствования технологической поддержки, поддерживающей технологию сопровождения и эксплуатации программ;
- создания новых, высокоинтеллектуальных средств инструментальной поддержки, состоящих из программных средств и средств вычислительной техники, обеспечивающих автоматизацию инженерных работ по моделированию, разработке и созданию самих программных систем.

Контрольные вопросы.

1. Перечислите задачи оценки эффективности разработки ИС
2. Что такое праксеология?
3. Основные праксеологические показатели.
4. Основные виды затрат.
5. Проблемы повышения эффективности ИС.

Тема 3. Оценка инженерной деятельности и продукта разработки.

Составляющие факторы оценки качества инженерной деятельности при проектировании ИС, структурные, функциональные, конструктивные критерии эффективности. Составляющие факторы оценки качества технологии разработки ИС. Основные свойства показателей качества.

Эффективность разработки ИС связана с поиском технически и экономически выгодных инженерных решений. *Продукт* современной инженерной деятельности должен быть конкурентоспособным. Инженерная деятельность должна быть рентабельна, выгодна, приносить прибыль, которая будет стимулировать инженерную и научную деятельность.

Модели представления проектных решений определяют совокупность понятий, привлекаемых для описания, как организации самого процесса инженерного проектирования, так и разрабатываемой информационной системы. Возникает проблема информационных технологий (ИТ) в оценке эффективности проводимых работ в разрезе как оценки качества технологии разработки ИС, так и оценки качества разрабатываемой ИС (рис.2.1). Рассмотрим пути решения этой проблемы с позиций парадигм апрагматической и прагматической методологий.



Рис.2.1. Оценка качества инженерной деятельности при разработке ИС.

1. Проблема эффективности оценки качества разрабатываемой информационной системы.

Данную проблему будем относить к апрагматическим проблемам, а соответствующие оценки качества - к разряду структурных критериев эффективности. Информационная система имеет единую цель функционирования - обработку информации, характеризуемую управляющими воздействиями, входными и выходными данными, алгоритмом обработки. Однако, являясь по своей сущности сложной системой, т.е. имея, иерархическую структуру с несколькими уровнями группирования и подчиненности отдельных модулей, каждый из которых имеет свою целевую задачу, ИС вызывает трудности с выявлением единого обобщенного критерия эффективности функционирования. Действительно, каждый отдельный компонент (модуль) системы имеет свой специфический частный критерий, как правило, не совпадающий и играющий подчиненную роль по отношению к критерию эффективности всей системы.

Еще одной проблемой является проблема выбора критерия эффективности при наличии обычно нескольких более или менее равнозначных критериев, каждый из которых может стать доминирующим в зависимости от внешних условий и состояния системы. Это обусловлено наличием внешней среды и тем, что каждая сложная система может являться подсистемой системы высшего уровня. Поэтому при разработке ИС приходится ограничиваться анализом критериев качества для некоторых типовых условий функционирования сложных систем, выделять необходимые для оптимизации параметры, оценивать влияние системы высшего уровня на выбор показателя качества.

Структурные критерии эффективности разделим на два вида: функциональные и конструктивные критерии качества.

– **Функциональные** критерии качества отражают уровень соответствия разработанной ИС ее целям функционирования, характеризуют особенности специфики применения ИС в соответствии с их целевым назначением (точность представления данных, диапазоны изменения параметров, время обработки, адаптивность к внешним воздействиям, достоверность результатов и др.). Функциональные критерии различны, и соответствуют разнообразию областей применения программных средств.

– **Конструктивные** критерии качества программных средств более или менее инвариантны к их целевому назначению и основным функциям (сложность программ; корректность программ, надежность функционирования; удобства доступа к данным и т.д.).

Отметим, что между функциональными и конструктивными критериями, как правило, прослеживается прямая зависимость и в ряде случаев функциональную оценку можно сопоставить с

соответствующей конструктивной (например, очевидно, что с увеличением количества элементов в системе (конструктивная сложность), уменьшается ее надежность при прочих равных условиях).

В некоторых случаях функциональные критерии можно свести к некоторым показателям обобщенной экономической эффективности применения программ в жизненном цикле. Эта экономическая эффективность может характеризоваться величиной экономии трудозатрат, энергии, материалов и т.д., выраженных через их стоимость или некоторый другой эквивалент. Эффективность функционирования ИС проявляется на этапе эксплуатации и возрастает по мере проведения модернизаций в процессе сопровождения.

2. Проблема эффективности проводимых работ в разрезе оценки качества технологии разработки ИС.

Информационные системы будем характеризовать как сложные системы с высокой стоимостью, с длительными сроками инженерного проектирования и изготовления. Таким образом, ИС следует рассматривать во всей полноте прагматических проблем, связанных с моделированием, исследованием, изготовлением, внедрением и эксплуатацией сложных систем. Такая характеристика ИС позволяет ставить проблему оценки затрат, при которой достигается та или иная эффективность технологии разработки. Степень эффективности всей технологической цепочки напрямую зависит от выбора методов решения функциональных задач и определяется конкретными инструментальными средствами и ресурсами для их реализации.

Технологические проблемы слабо связаны с функциональным назначением проектируемой системы, поэтому методы их решения могут рассматриваться отдельно от конкретного целевого назначения ИС.

Длительность и трудоемкость изготовления ИС большого объема может оказаться определяющей для затрат и сроков создания всей информационной системы. В этом случае длительность их разработки определяет качество и степень автоматизации технологии разработки ИС, а в конечном итоге и качество самой ИС.

В настоящее время отсутствует целостная методика выбора характеристик качества ИС и способов их оценки (или измерения), и поэтому все результаты в данной области должны рассматриваться скорее как информация к размышлению, чем как окончательные выводы или предписания. Будем использовать в концепции ИТ наиболее рациональный способ действий по оценке качества ИС, который на сегодняшний день состоит в том, чтобы разработать некоторую систему показателей качества (полезности) и использовать ее для определения направления дальнейшего усовершенствования ИС. Определим основные задачи, на решение которых будем делать акцент при разработке системы показателей полезности. В идеале, показатели полезности должны позволить:

- сравнивать различные варианты архитектурной компоновки моделей представления ИС;
- анализировать причины нежелательного отклонения показателей от желаемых при учете огромного числа формальных и неформальных ограничений на компоновку моделей представления;
- отражать обобщенную «полезность» для общества технологий проектирования программных средств, которая в первую очередь характеризуется трудоемкостью и длительностью создания;
- оценивать качество программ, достигаемое при применении соответствующих технологий изготовления ИС.

В основу оценки эффективности создания ИС положены процессы анализа показателей качества программных средств, а также технико-экономических показателей их цикла разработки.

Очертим круг правил, которым должны удовлетворять показатели технологии разработки в идеале:

- численно и в наиболее общем виде характеризовать степень выполнения системой своей основной целевой функции;
- позволять выявить и оценить степень влияния на эффективность системы различных факторов и параметров и в том числе затрат различного вида на ее реализацию;
- быть простым и иметь малую дисперсию, т.е. слабо зависеть от случайных неконтролируемых факторов.

Трудно вывести обобщенный критерий, связывающий физические зависимости с величинами, характеризующими ту цель, которой в своем действии должен служить разрабатываемый объект.

Контрольные вопросы.

1. Поясните понятие технически и экономически выгодных инженерных решений.
2. Структурные критерии эффективности и их два вида.
3. Технологические критерии эффективности.
4. Какими свойствами должны обладать показатели полезности?

Тема 4. Планирование работ по этапам и стадиям проектирования.

Предпроектные работы. Структура этапов и стадий проектирования. Разработка план-графиков, диаграмм сроков выполнения. Учет капитальных и эксплуатационных затрат.

Для поддержки жизнеспособности любого промышленного, в том числе и программно-технического изделия необходимы постоянные затраты. С точки зрения инженерного процесса, зависимость необходимых затрат в увязке с жизненным циклом разрабатываемой ИС выглядит примерно, так как представлено на рисунке 1. Всплеск затрат на начальных этапах присущ стадиям проектирования и разработки, период минимальных затрат – этапам сопровождения и эксплуатации, сильное увеличение затрат характеризует период старения системы.

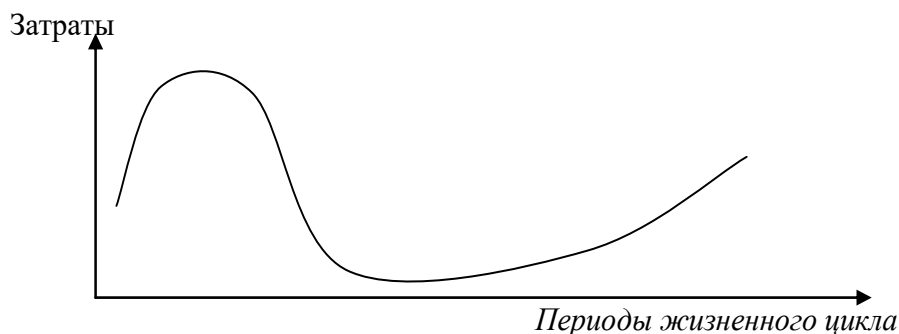


Рис.1. График соответствует зависимости затрат от жизненного цикла системы.

Остановимся на основных этапах инженерного проектирования – стадиях разработки.

Стадии разработки определяют наиболее общий состав процедур разработки и требования к документации. Стадии разработки регламентируются ГОСТом и другими нормативными документами.

Выделим четыре стадии разработки: техническое задание, эскизный проект, технический проект, рабочий проект. Проектирование ИС на ранних стадиях характеризуется высокой неопределенностью исходных данных и представлений разработчиков о свойствах и функциях создаваемой системы.

Уровни абстрагирования определяют систему понятий (модель абстракции), привлекаемых для описания инженерных решений. Уровни представления определяются в рамках конкретной предметной области, методики моделирования, могут регламентироваться различными стандартами. Уровень абстрагирования будем сопоставлять с видом моделей абстракций, а переход по уровням абстрагирования – с переходом на новый вид модели абстракций.

Уровни детализации определяют степень детализации элементов и связей компонент моделируемой системы при описании ее на одном уровне абстрагирования. Уровни детализации могут регламентироваться конкретными методиками моделирования. Под уровнем детализации будем понимать соответствующий иерархический уровень в модели абстракций (модель абстракций – иерархическая структура).

Уровни определенности характеризуют форму описания моделей. Наиболее существенными уровнями определенности являются концептуальный уровень, логический уровень и физический уровень.

– Концептуальный уровень – содержательное описание модели исходя из содержательного процесса управления. Характеризуется неформальными (слабо формализованными) средствами описания инженерных решений.

– Логический уровень – представление моделей системы с использованием типового математического аппарата, на основе которого можно проводить анализ и синтез структур и адекватно отображать с заданной степенью соответствия реальные процессы. Характеризуется формально обоснованными инженерными решениями.

– Физический уровень – описание модели системы на уровне программно-аппаратных средств реализации. Характеризуется практической выполнимостью моделей абстракций на программном уровне.

В таблице 1. Представлен «примитив» план-графика с предложенными видами работ по этапам для разработки программно-технического средства.

Таблица 1. План-график (пример этапов и видов работ разработки ИС).

ПТС – программно-технические средства; ПО – программное обеспечение.

№ пп	Название этапов и видов работ	Длительность выполнения (кал. месяц)	Стоимость (уе)	Материалы и изделия, предоставляемые Заказчику
1.	Предпроектная подготовка	2	13182	
1.1	<u>Разработка технического задания.</u> 1. Сбор материалов для формирования исходных данных для планирования и проектирования ИС. 2. Техничко-экономическое обоснование. 3. Обоснование проведения научных исследований. 4. Определение требований к ИС, стадиям, этапам и срокам разработки. 5. Оформление технического задания	1		Документы. • Техническое задание • План-график • Калькуляция
1.2	<u>Системный анализ предметной области и класса проектируемой ИС.</u> 1. Определение целей и назначения ИС; 2. Проектирование и моделирование основных функций и обобщенных алгоритмов. 3. Выбор методов решения задач. 4. Выбор и обоснование критериев эффективности и качества разработки ИС.	1		Документы. • Исходные данные для проектирования. • Сценарии. • Иерархия функций. • Топология ИС. • Конфигурация ПО и ТС. • Обобщенные алгоритмы. • Список показателей и критериев эффективности и качества
2.	Эскизный проект	2	20370	
2.1	<u>Проектирование архитектуры ИС.</u> 1. Формирование общей структуры ИС и её основных компонент: определение структуры ПО, определение структуры модулей ПО, ТС. 2. Распределение ресурсов ТС по функциональным задачам ПО: 3. Оценка производительности ПТС: распределение емкостей накопителей информации и памяти ЭВМ, пропускной способности коммутаторов и каналов связи и пр. 4. Формирование дисциплины взаимодействия процессоров и диспетчеризации вызова программ.	1		Документация. • Покомпонентная спецификация ИС. • Оценка и распределение ресурсов ТС по компонентам ПО. • Инструкции по составлению спецификаций на модули и группы программ. • Методика отладки и комплексирования программ. • Спецификация взаимодействия параллельных задач и диспетчеризация модулей.
2.2	<u>Подготовка технологических средств.</u> 1. Организация базы данных проекта ИС; 2. Выбор и адаптация инструментария и языков программирования, настройка средств трансляции и отладки; 3. Оценка реализуемости данного класса ПО на базе выбранных ТС. 4. Выбор или разработка инструкций по применению технологии проектирования-конструирования ПТС.	1		Документы. • Обоснование выбора инструментальных средств программирования. • Структура технологического процесса разработки ПТС. • Формальная структура, информационная, функциональная, потоковая. • Методика конструирования ИС
3.	Технический проект	5	81030	

3.1	Разработка ПО 1. Разработка алгоритмов, спецификаций на модули и группы программ. 2. Конструирование информационного фонда (базы данных) 3. Программирование и трансляция ПО.	3		Документы. • Внешние спецификации модулей • Логика модулей (определение данных, алгоритм, программа) • Структура сопряжений модулей • Глобальная модель данных + локальные модели данных. Изделия • Исходный модуль версии ПО, загрузочные (исполнимые) модули на отдельных магнитных или оптических носителях.
3.2	Отладка программ в статике. 1. Планирование отладки программ. 2. Тестирование программ. 3. Локализация ошибок и корректировка программ. 4. Комплексирование программ.	1		Документы • Методика детерминированного тестирования: тест, исходные данные и эталонные результаты.
3.3	Комплексная динамическая отладка: 1. Выбор средств для имитации абонентов. 2. Разработка программ имитации. 3. Создание программ обработки результатов. 4. Отладка функционирования ПО в реальном масштабе времени. 5. Отладка программы на объектах Заказчика.	1		Документы. • Структура и спецификация модели объекта автоматизации. Изделия • Программные и аппаратные имитаторы внешней среды. • Средства для контроля и регистрации промежуточных данных, облегчающие обнаружение и локализацию ошибок • Исходный модуль версии ПО, загрузочные (исполнимые) модули на отдельных магнитных или оптических носителях.
4.	Рабочий проект	9	169328	
4.1	Испытания ПО. 1. Разработка, согласование и утверждение программы и методики испытаний ПО: испытания на полноту функционирования; испытания на надежность функционирования и другие характеристики. 2. Обработка результатов испытаний. 3. Разработка акта испытания. 4. Проведение корректировки ПО и программной документации по результатам испытаний.	3		Документы. • Программа испытаний. • Акт испытаний.
4.2	Выпуск машинных носителей и документирование. 1. Разработка инсталляционной версии пакета ПО с приданием ему статуса Программного Продукта (ПП). 2. Изготовление машинных носителей и выпуск тиража.	1		Изделия. • Макет обложки • Инсталляционная версия ПО, загрузочные (исполнимые) модули на отдельных магнитных или оптических носителях. • Тираж ПО в составе: инсталляционная версия (CD), пакет документов (брошюры)
4.3	Разработка и изготовление программной документации. 1. Эксплуатационных документов. 2. Технологических документов. 3. Исследовательских документов.	5		Документы • Руководство пользователя • Руководство программиста • Проектно-конструкторская документация. • Акт о закрытии договора разработки Проработка новых юридических документов на внедрение и сопровождение ИС
5.	Внедрение и сопровождение			

На рисунке 2. изображён пример наглядной диаграммы со сроками выполнения этапов

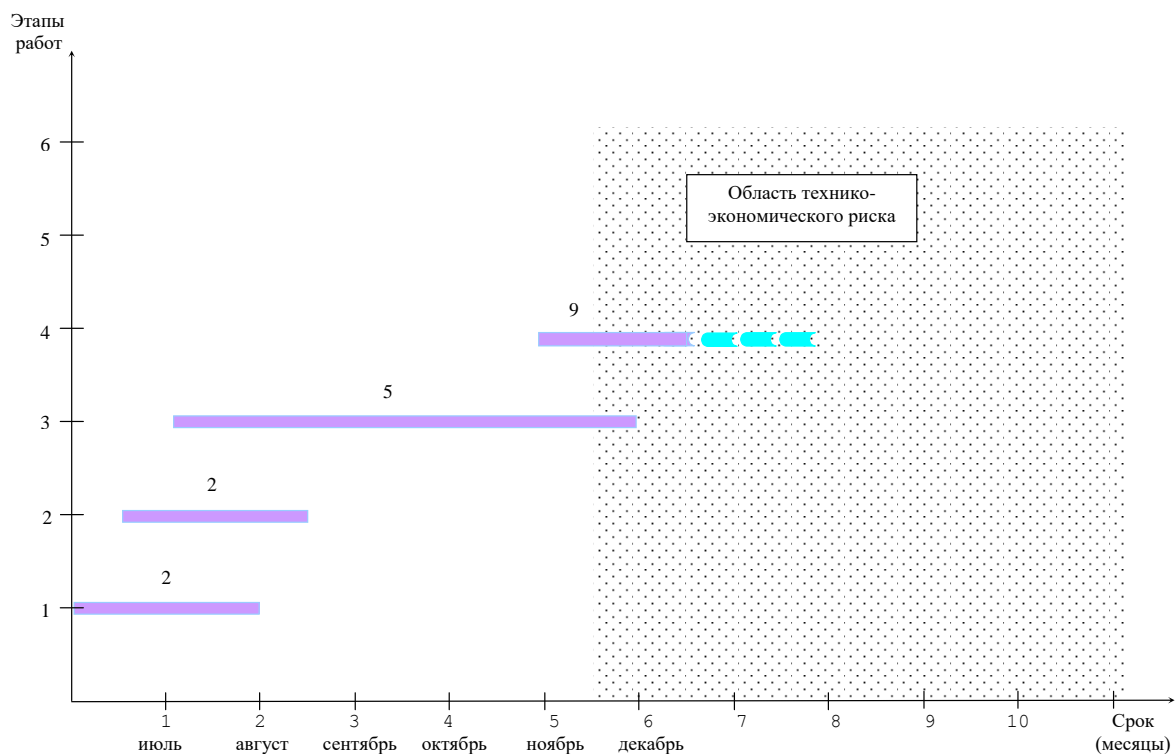


Рис.2. Диаграмма сроков выполнения работ

Таблица 2. Общая стоимость работ составила:

№ этапа	Стоимость
1. Предпроектная подготовка	13182
2. Эскизный проект	23070
3. Технический проект	81030
4. Рабочий проект	169328
ИТОГО	286610

Таблица 3 Пример учёта эксплуатационных и капитальных затрат

Наименование затрат	Стоимость
1. Человеко-машинный комплекс ЧМК (кол-во ИТП×стоим.чм)	4000
2. Специализированные технические средства (СТС)	0
3. Вспомогательное оборудование и инвентарь (10%ЧМК)	400
4. Транспортировка, монтаж и наладка СТС (10%СТС)	0
5. Производственно-хозяйственный инвентарь и прочее оборудование 1,5%ЧМК)	60
6. Расходные материалы (бумага, диски, картриджи и др.) (2%ЧМК)	80
7. Электроэнергия ЧМК и СТС (1%(ЧМК+ СТС))	40
8. Ремонты или «больничные» ЧМК (5%ЧМК)	200
9. Освещение, отопление, уборку и охрану помещений (0,5%ЧМК)	20
10. Стоимость услуг сторонних организаций (1%ЧМК),	40
11. Содержание транспорта (1%ЧМК),	40
12. Аппарат управления (10%ЧМК)	400
13. Накладные и непредвиденные расходы (1%ЧМК)	40
ИТОГО:	5320

Контрольные вопросы

1. Зависимость затрат от жизненного цикла системы.
2. Стадии проектирования.
3. Что такое концептуальный, логический и физический уровень описания.

4. Структура, форма и содержание план-графика.
5. Смысл диаграмм сроков выполнения.
6. Отличие эксплуатационных и капитальных затрат.

Тема 5. Оценка экономической эффективности автоматизированной системы управления предприятием.

Экономическая эффективность АСУП и факторы, ее определяющие. Расчет годового объема реализуемой продукции. Расчет изменения себестоимости продукции предприятия. Расчет единовременных затрат на создание и внедрение АСУП.

Оценка экономической эффективности автоматизированной системы управления предприятием

I. Основные положения

Автоматизированная система управления предприятием создается для повышения эффективности производственно-хозяйственной деятельности предприятия. Этот эффект достигается за счет повышения качества управления предприятием, так как в условиях функционирования АСУП полностью или частично автоматизируется процесс решения ряда задач управления, используется интегрированная информационная база данных, повышается качество и оперативность принимаемых управленческих решений.

Экономическая эффективность АСУП и факторы, ее определяющие

Факторы, вызывающие экономический эффект

1. Совершенствование механизма управления
2. Оперативность и достоверность получаемой информации
3. Рациональное использование всех видов производственных ресурсов
4. Сокращение потерь материальных ресурсов
5. Улучшение использования рабочего времени
6. Повышение уровня использования средств производства
7. Обеспечение ритмичности производства
8. Изменение характера и устава труда управленческого персонала

Составляющие экономического эффекта

1. Рост объема реализуемой продукции
2. Повышение уровня качества выпускаемой продукции
3. Увеличение фондоотдачи
4. Ускорение оборачиваемости
5. Рост производительности труда
6. Сокращение непроизводительных расходов
7. Снижение затрат на рубль выпущенной продукции
8. Увеличение прибыли и повышение рентабельности производства

Экономическая эффективность АСУП оценивается с использованием двух хозрасчетных показателей: годового прироста прибыли (годовой экономии) и коэффициента экономической эффективности капитальных вложений.

В качестве базы для сравнения при определении годового прироста прибыли принимаются показатели производственно-хозяйственной деятельности предприятия (без учета влияния АСУП), планируемые на год, следующий за годом ввода АСУП в эксплуатацию, если АСУП внедряется на действующем предприятии. Если АСУП разрабатывается для проектируемого предприятия, то в качестве базы для сравнения принимаются фактические показатели предприятия-аналога с наименьшей величиной приведенных затрат на рубль реализуемой продукции.

Годовой прирост прибыли $\Delta\Pi$ рассчитывается по формуле:

$$\Delta\Pi = \Pi_1 \cdot (A_1 - A_2) / A_1 + A_2 \cdot (C_1 - C_2) / 100 \quad (1)$$

где A_1, A_2 - годовой объем реализуемой продукции до и после внедрения АСУП соответственно, тыс.р; Π_1 - прибыль от реализации продукции до внедрения АСУП, тыс. р.; C_1, C_2 - затраты на рубль реализуемой продукции до и после внедрения АСУП соответственно, к.

В условиях функционирования АСУП возможна дополнительная прибыль за счет сокращения непроизводственных расходов (оплата штрафов, пени, неустоек) вследствие более действенного контроля финансовой деятельности предприятия. Дополнительная прибыль возможна также за счет повышения качества продукции предприятия, если оно не находит отражения в надбавках к оптовым ценам. Рассматриваемая прибыль учитывается в формуле (1) дополнительно.

Вторым хозяйственным показателем эффективности АСУП является коэффициент экономической эффективности капитальных вложений в АСУП E . Он рассчитывается по формуле:

$$E = \Delta\Pi / K_k, E \geq E_{\text{нор}} \quad (2)$$

где K_k - капитальные вложения на создание АСУП, тыс. р.; $E_{\text{нор}}$ ~ нормативный коэффициент эффективности капитальных вложений на создание АСУП.

Расчетный коэффициент эффективности E сравнивается с отраслевым нормативным значением $E_{\text{нор}}$. Если неравенство (2) выполняется, то АСУП считается эффективной.

2. Расчет годового объема реализуемой продукции

Внедрение АСУП позволяет улучшить использование производственных ресурсов, к числу которых относятся сырье, материалы, комплектующие изделия, труд, станки и оборудование, топливо и энергия, здания и сооружения. Улучшение использования ресурсов ведет к их экономии. Вовлечение сэкономленных производственных ресурсов в производственный процесс обеспечивает прирост объема реализуемой продукции. Таким образом, в результате внедрения АСУП объем реализуемой продукции возрастает от A_1 до величины A_2 .

Количественная оценка прироста объема реализации возможна на основе анализа вклада в экономию производственных ресурсов каждой задачи управления, входящей в состав АСУП. Если в результате решения i -ой задачи управления в АСУП обеспечивается экономия j -го производственного ресурса, то в процессе эксплуатации этой задачи в течение года обеспечивается экономия ΔP_{ij} рассматриваемого ресурса. Величина общей экономии j -го ресурса за счет функционирования АСУП ΔP_j определяется как сумма соответствующих величин по всем задачам АСУП:

$$\Delta P_j = \sum_{i=1}^m \Delta P_{ij}, j=1..n \quad (3)$$

где m - количество задач, решаемых в АСУП; n - количество видов производственных ресурсов.

Использование каждого резерва $\Delta P_j, j=1..n$ потенциально ведет к приросту объема реализуемой продукции предприятия на величину:

$$\Delta A_j = \Delta P_j / q_j \quad (4)$$

где q_j - удельный расход j -го производственного ресурса на рубль или единицу выпускаемой продукции.

Однако на практике производственные ресурсы взаимосвязаны и для обеспечения прироста объема реализации на величину ΔA_j за счет j -го ресурса требуется наличие определенных резервов других ресурсов $\Delta P_k, k \neq j$. Например, экономия материалов сама по себе не ведет к росту реализации продукции предприятия. Для использования этого резерва в производстве требуется наличие резерва трудовых ресурсов, резервов технологического оборудования и т.д.

Таким образом, величина прироста объема реализуемой продукции ΔA сложным образом связана с резервами производственных ресурсов ΔP_j и может быть определена в результате решения оптимизационной задачи

$$\Delta A \rightarrow \max \text{ при } \Delta P'_j \leq \Delta P_j, j=1..n \quad (5)$$

где $\Delta P'_j$ - то количество j -го ресурса, которое необходимо для выпуска продукции ΔA .

Величина резерва j -го ресурса определяется выражением (3), а связь между $\Delta P'_j$ и ΔA устанавливается формулой (4).

В задаче (5) первая строка является критерием оптимальности, а вторая строка представляет собой систему ограничений. Искомой в задаче (5) является максимальная величина прироста объема

реализации ΔA , которая может быть достигнута за счет использования экономии производственных ресурсов ΔP_j , $j=1..n$ получаемой в результате функционирования АСУП.

Годовой объем реализуемой продукции предприятия после внедрения АСУП определяется выражением

$$A_2 = A_1 + \Delta A \quad (6)$$

3. Расчет изменения себестоимости продукции предприятия

Наряду с обеспечением роста объема реализуемой продукции внедрение АСУП ведет к снижению себестоимости продукции предприятия. Величина себестоимости годового выпуска реализуемой продукции до внедрения АСУП определяется выражением:

$$C = C_m + C_t + C_z + C_{бр} + C_{п} + C_{об} + C_{ц} + C_o + C_{вн} + C_{ппр} , \quad (7)$$

где C_m - затраты на сырье и материалы за вычетом возвратных расходов, тыс. р.; C_t - затраты на топливо и энергию на технологические цели, тыс. р.; C_z - основная и дополнительная заработная плата производственных рабочих с отчислениями на социальное страхование, тыс. р.; $C_{бр}$ - потери от брака, тыс. р.; $C_{п}$ - расходы на подготовку и освоение производства, тыс. р.; $C_{об}$ - расходы на содержание и эксплуатацию оборудования, тыс. р.; $C_{ц}$ - цеховые расходы, тыс. р.; C_o - общезаводские расходы, тыс. р.; $C_{вн}$ - внепроизводственные расходы, тыс.р.; $C_{ппр}$ - прочие производственные расходы, тыс. р.

В результате внедрения АСУП изменяются перечисленные в формуле (7) элементы затрат, что ведет к снижению себестоимости продукции предприятия на величину

$$\Delta C = \sum_{i=1}^{10} \Delta C_i , \quad (8)$$

где ΔC_i - снижение величины i -го элемента затрат, тыс. р.; i - номер элемента затрат в формуле (7).

Вместе с тем при переходе к АСУП появляются дополнительные расходы $C_{экс}$, связанные с эксплуатацией АСУП.

Затраты на рубль реализуемой продукции C_1 и C_2 до и после внедрения АСУП [см формулу (I)] при известных значениях показателей C , A_1 и A_2 определяются выражениями:

$$C_1 = \frac{C}{A_1} * 100 , \quad (9)$$

$$C_2 = \frac{C - \Delta C - C_{экс}}{A_2} * 100 , \quad (10)$$

Расчет величины ΔC осуществляется последовательно по статьям затрат формулы (7). Расчет экономии затрат ΔC_m на сырье и материалы связан с экономией ресурса «сырье и материалы» в результате функционирования АСУП. Как показано в гл.2, §2 величина экономии данного ресурса ΔP , определяемая формулой (3), частично используется для дополнительного выпуска продукции ΔA (5).

Разница между величиной резерва ресурса «сырье и материалы» ΔP и количеством этого ресурса и $\Delta P'$ необходимым для выпуска продукции ΔA , представляет собой величину излишнего высвобожденного в результате функционирования АСУП производственного ресурса. Этот ресурс может быть реализован предприятием или передан другому, родственному предприятию. В результате получаем экономию затрат

$$\Delta C_m = \sum_{k=1}^l (\Delta P_k - \Delta P'_k) , \quad (11)$$

где k - номер вида используемых на предприятии сырья и материалов; l - общее количество видов сырья и материалов.

Величина экономии затрат на топливо и энергию на технологические цели ΔC_t рассчитывается по видам используемых на предприятии топлива и энергии аналогично расчету экономии затрат на сырье и материалы по формуле (11).

Экономия по фонду заработной платы производственных рабочих с отчислениями на социальное страхование ΔC_3 в условиях функционирования АСУП определяется по формуле:

$$\Delta C_3 = \Delta C_{3п} + \Delta C_{3о} + \Delta C_{3с} \quad , \quad (12)$$

где $\Delta C_{3п}$ – экономия по фонду заработной платы, образующаяся за счет уменьшения потерь рабочего времени в результате функционирования АСУП, тыс.р.; $\Delta C_{3о}$ – условная экономия, образующаяся за счет опережения темпов роста производительности труда по сравнению с ростом заработной платы, тыс.р.; $\Delta C_{3с}$ – экономия по фонду заработной платы производственных рабочих в результате сокращения доплат за сверхурочные работы при функционировании АСУД, тыс.р.

Значением $\Delta C_{3п}$ определяется по формуле (11) по всем категориям основных производственных рабочих: профессиям и специальностям, квалификации, системам оплаты труда. Условная экономия $\Delta C_{3о}$ рассчитывается по формуле:

$$\Delta C_{3о} = C_3 (\gamma - 1)(1 - \alpha) \quad (13)$$

где C_3 - заработная плата производственных рабочих без учета влияния АСУП, тыс.р.; γ - коэффициент роста объема реализуемой продукции, равный A_2/A_1 ; α - коэффициент зависимости темпов прироста средней заработной платы от темпов прироста производительности труда, равный $T_{3т}/T_{пт}$ ($T_{3т}$ - темп прироста средней заработной платы за t -летний период, %; $T_{пт}$ - темп прироста производительности труда за t -летний период, %).

Для расчета коэффициента α берутся данные за последние t лет (например $t=3$) по объему товарной (валовой) продукции B_t , фонду заработной платы $C_{3т}$, средней численности производственных рабочих $Ч_t$.

Темп прироста средней заработной платы рассчитывается по формуле

$$T_{3т} = \left(\sqrt[t]{\frac{C_{3т} * Ч_1}{Ч_t * C_{31}}} - 1 \right) * 100 \quad , \quad (14)$$

где $C_{3т}$ - фонд заработной платы в году внедрения АСУП без учета ее влияния (в t -м году рассматриваемого периода), тыс.р.; C_{31} - фонд заработной платы за первый анализируемый год, тыс.р.

Темп прироста производительности труда определяется формулой:

$$T_{пт} = \left(\sqrt[t]{\frac{B_t * Ч_1}{Ч_t * B_1}} - 1 \right) * 100 \quad , \quad (15)$$

где B_t – объем товарной (валовой) продукции в год внедрения АСУП без учета ее влияния, тыс.р.; B_1 – объем товарной (валовой) продукции за первый анализируемый год, тыс.руб.

Экономия от сокращения доплат за сверхурочные работы, образуется в результате повышения ритмичности производства в условиях функционирования АСУП.

Величина экономии $\Delta C_{бр}$, получаемая на предприятии после внедрения АСУП в результате снижения потерь от брака, определяется по формуле;

$$\Delta C_{бр} = (h_1 - h_2) * C / 100 \quad C_{бр} * (\gamma - 1) * \beta_{бр} \quad (16)$$

где h_1 , h_2 - процент брака на предприятии до и после внедрения АСУП, $h = 100 * (C_{бр} / C)$; C - себестоимость продукции до внедрения АСУП, тыс. р.; $C_{бр}$ - потери от брака, тыс. р.; γ - коэффициент роста объема реализуемой продукции; $\beta_{бр}$ - коэффициент зависимости прироста потерь от брака при увеличении объема производства.

Первая часть формулы позволяет оценить величину эффекта, получаемого от снижения процента брака в производстве в результате внедрения АСУП. Вторая часть - определяет дополнительные потери от брака, обусловленные ростом объема производства.

Экономия затрат на подготовку и освоение производства $\Delta C_{п}$ в условиях функционирования АСУП связана с сокращением расходов на проектирование и конструирование новых изделий, на разработку технологического процесса их изготовления, на перестановку и переналадку технологического оборудования. Значение показателя $\Delta C_{п}$ определяется прямым счетом.

Расходы на содержание и эксплуатацию оборудования $C_{об}$ состоят из условно-переменных и условно-постоянных затрат. Условно-переменная составляющая изменяется пропорционально росту объема производства, в то время как условно-постоянная составляющая не зависит от данного фактора.

В условиях функционирования АСУП имеет место изменение объема производства, следовательно, величина экономии

$$\Delta C_{об} = C_{обп} * (1 - \gamma) \quad (17)$$

где $C_{обп}$ - условно-переменная составляющая, тыс.р.; γ - коэффициент роста объема реализуемой продукции.

Величина экономии цеховых расходов предприятия $\Delta C_{ц}$, рассчитывается по формуле:

$$\Delta C_{ц} = \Delta C_{цз} + \Delta C_{цн} - \Delta C_{ца} \quad (18)$$

где $\Delta C_{цз}$ - экономия по фонду заработной платы аппарата управления цехов, возникающая в условиях функционирования АСУП вследствие централизации обработки на вычислительном центре цеховой документации, тыс.р.; $\Delta C_{цн}$ - снижение цеховых непроизводительных расходов в условиях функционирования АСУП, тыс.р.; $\Delta C_{ца}$ - дополнительные цеховые расходы, связанные с ростом объема производства, тыс.р.

Экономия по фонду заработной платы цехового персонала рассчитывается по формуле:

$$\Delta C_{цз} = \Delta \text{Ч}_{ц} * Z_{ц} * (1 + H_{д}) * (1 + H_{с}) / 1000, \quad (19)$$

где $\Delta \text{Ч}_{ц}$ - численность высвобожденных работников аппарата управления цехов, а также инженерно-технических и других работников, не относящихся к управленческому персоналу, чел.; $Z_{ц}$ - средняя заработная плата (основная) одного высвобожденного работника аппарата управления цехов, р.; $H_{д}$, $H_{с}$ - коэффициенты, учитывающие дополнительную заработную плату и отчисления на социальное страхование соответственно.

Экономия от сокращения непроизводительных расходов $\Delta C_{цн}$ в условиях функционирования АСУП обусловлена снижением потерь от простоев (выплаты рабочим за время простоев, доплаты за отвлечение работников на менее квалифицированную работу), снижением потерь от порчи материальных ценностей при хранении их в цехах, от недоиспользования деталей, узлов, технологической оснастки. Значение

$$\Delta C_{цн} = \Delta C_{цнп} - C_{цн} * (\gamma - 1) * \beta_{цн}, \quad (20)$$

где $\Delta C_{цнп}$ - снижение цеховых непроизводительно расходов в условиях функционирования АСУП, тыс.р.; $C_{цн}$ - непроизводительные расходы в составе цеховых расходов до внедрения АСУП, тыс.р.; γ - коэффициент роста объема реализуемой продукции;

$\beta_{цн}$ - коэффициент зависимости прироста непроизводительных цеховых расходов от прироста объема производства.

Дополнительные цеховые расходы $\Delta C_{ца}$ рассчитываются по формуле:

$$\Delta C_{ца} = C_{ц} * (\gamma - 1) * \beta_{ц}, \quad (21)$$

где $C_{ц}$ - цеховые расходы до внедрения АСУП, тыс. р.; $\beta_{ц}$ - коэффициент зависимости прироста цеховых расходов от прироста объема производства.

Величина сокращения общезаводских расходов $\Delta C_{о}$ рассчитывается аналогично экономии цеховых расходов:

$$\Delta C_{о} = \Delta C_{оз} + \Delta C_{он} - \Delta C_{оа}, \quad (22)$$

где $\Delta C_{оз}$ - экономия по фонду заработной платы аппарата управления, возникающая в результате совершенствования структуры управления в связи с внедрением АСУП, тыс. р.; $\Delta C_{он}$ - сокращение расходов на управление предприятием (кроме заработной платы), общезаводских, а также непроизводительных общезаводских расходов в связи с внедрением АСУП, тыс. р.; $\Delta C_{оа}$ - дополнительные общезаводские расходы, связанные с ростом объема производства, тыс. р.

Значения $\Delta C_{оз}$, $\Delta C_{он}$, $\Delta C_{оа}$ определяются по формулам:

$$\Delta C_{оз} = \Delta \text{Ч}_{о} Z_{о} (1 + H_{д}) (1 + H_{с}) 10^{-3}; \quad (23)$$

$$\Delta C_{он} = \Delta C_{онп} - C_{он} (\gamma - 1) \beta_{он}; \quad (24)$$

$$\Delta C_{оа} = C_{о} (\gamma - 1) \beta_{о}; \quad (25)$$

где $\Delta \text{Ч}_{о}$ - численность высвобожденных работников аппарата управления в условиях функционирования АСУП, чел.; $Z_{о}$ - средняя заработная плата (основная) одного высвобожденного работника аппарата управления, р.; $\Delta C_{онп}$ - снижение общезаводских непроизводительных расходов, тыс.р.; $C_{он}$ - непроизводительные расходы в составе общезаводских расходов до внедрения АСУП, тыс.р.; $\beta_{он}$ - коэффициент зависимости прироста непроизводительных общезаводских расходов от

прироста объема производства; C_o - общезаводские расходы до внедрения АСУП, тыс.р.; β_o - коэффициент зависимости прироста общезаводских расходов от прироста объема производства.

Величина внепроизводственных расходов $C_{вн}$ на тару и упаковку продукции предприятия, на погрузку продукции в транспортные средства и доставку ее на станцию отправления потребителю изменяется пропорционально объему производства и, следовательно, определяется выражением:

$$\Delta C_{вн} = C_{вн}(1-\gamma) \quad (26)$$

Прочие производственные расходы $C_{ппр}$ после внедрения АСУП остаются практически неизменными и поэтому $\Delta C_{ппр} = 0$.

После внедрения АСУ себестоимость годового объема реализуемой продукции предприятия изменяется на величину текущих затрат $C_{экс}$, связанных с эксплуатацией АСУП. Значение этих затрат, тыс.р., рассчитывается по формуле:

$$C_{экс} = C_{ивц} + C_T + C_o + C_n + C_э, \quad (27)$$

где $C_{ивц}$ - затраты на эксплуатацию информационно-вычислительного центра, тыс.р.; C_T , C_o - заработная плата технического и обслуживающего персонала информационно-вычислительного центра соответственно, тыс.р.; C_n - накладные расходы информационно-вычислительного центра, тыс.р.; $C_э$ - затраты на электроэнергию информационно-вычислительного центра, тыс.р. Величины C_T и C_o рассчитываются по формулам:

$$C_T = Z_T(1+N_d)(1+N_c)10^{-3}; \quad (28)$$

$$C_o = Z_o(1+N_d)(1+N_c)10^{-3}; \quad (29)$$

где Z_T , Z_o - средняя заработная плата (основная) одного работника информационно-вычислительного центра из числа технического и обслуживающего персонала соответственно, р.

Накладные расходы информационно-вычислительного центра составляют

$$C_n = [Z_T(1+N_d) + Z_o(1+N_c)] N_n 10^{-3}, \quad (30)$$

где N_n - коэффициент накладных расходов.

Затраты на электроэнергию определяются выражением

$$C_э = (\Phi_n - t_{рем}) N_n \Pi_э 10^{-3}, \quad (31)$$

где Φ_n - номинальный годовой фонд времени работы ЭВМ, ч; $t_{рем}$ - время на планово-предупредительные ремонты и прочие плановые простои, ч; N_n - установленная мощность оборудования, кВт; $\Pi_э$ - стоимость 1 кВт.ч электроэнергии, р.

4. Расчет единовременных затрат на создание и внедрение АСУП

Единовременные затраты k на создание и внедрение АСУП рассчитываются по формуле;

$$k = k_n + k_k - \Delta O, \quad (32)$$

где k_n - предпроизводственные затраты, представляющие собой единовременные расходы на разработку проекта АСУП, тыс. р.; k_k - капитальные вложения на создание АСУП, тыс. р.; ΔO - сокращение величины оборотных средств в условиях функционирования АСУП, тыс. р.

В состав предпроизводственных затрат k_n , включаются расходы на научно-исследовательские и проектные работы по созданию АСУП, включая разработку алгоритмов и программ, составление инструкций, справочников и других документов по эксплуатации АСУП, подготовку и переподготовку кадров. При создании АСУП на базе разработанных ранее проектов прикладных программ или типовых проектных решений в предпроизводственные затраты включаются лишь расходы на их доработку и привязку к условиям конкретного предприятия. Величина предпроизводственных затрат определяется на основе действующих методик расчета сметы затрат на научно-исследовательские и опытно-конструкторские работы.

Капитальные вложения k_k на создание АСУП включают затраты на приобретение вычислительной техники, периферийных устройств, средств связи, вспомогательного оборудования и прочей оргтехники и хозяйственного инвентаря, а также на строительство (реконструкцию) зданий и сооружений, необходимых для функционирования АСУП. Капитальные вложения включают также остаточную стоимость высвобождаемого (ликвидируемого) оборудования, устройств, зданий, сооружений $k_{кв}$ рассчитываемую по формуле:

$$k_{кв} = \sum_{i=1}^n k_{вi} \left(1 - \frac{H_A T_{эi}}{100} \right), \quad (33)$$

где $k_{вi}$ - первоначальная стоимость высвобождаемых (ликвидируемых) основных фондов i -го вида, тыс.р.; H_A - годовая норма амортизации, %; $T_{эi}$ - длительность эксплуатации основных фондов i -го вида, год.

Сокращение величины оборотных средств ΔO определяется по формуле:

$$\Delta O = (\Delta T_m C_m + \Delta T_{ц} V + \Delta T_c A_1) / 360, \quad (34)$$

где ΔT_m - сокращение времени нахождения сырья, материалов, полуфабрикатов на складах предприятия в условиях АСУП, сутки; C_m - годовая стоимость сырья, материалов, полуфабрикатов предприятия, тыс.р.; $\Delta T_{ц}$ - сокращение длительности производственного цикла предприятия в условиях функционирования АСУП, сутки; V - годовой объем валовой продукции, тыс.р.; ΔT_c - сокращение времени хранения готовой продукции на складах, сутки; A_1 - годовой объем реализуемой продукции, тыс.р.

Значения ΔT_m , $\Delta T_{ц}$, ΔT_c , определяются на основе анализа производственно-хозяйственной деятельности предприятия. При расчете экономической эффективности использования отдельной зад. ли АСУП капитальные затраты k_k и сокращение величины оборотных средств ΔO учитываются пропорционально доле машинного времени, используемого при решении данной задачи в процессе функционирования АСУП. Предпроизводственные затраты k_n принимаются пропорционально доле времени на разработку, отладку и внедрение данной задачи из общего времени на разработку задач АСУП. Определение экономической эффективности АСУП осуществляется с учетом фактора времени, если капитальные вложения на создание и внедрение АСУП реализуются в течение ряда лет. Фактор времени учитывается путем приведения капитальных вложений на создание и внедрение АСУП и текущих затрат на эксплуатацию АСУП к одному моменту времени (началу расчетного года). Такое приведение выполняется умножением (делением) затрат и результатов соответствующего года на коэффициент приведения α_t , определяемый по формуле:

$$\alpha_t = (1+E)^t, \quad (35)$$

где E - норматив приведения, равный 0,1; t - число лет, отделяющее затраты и результаты рассматриваемого года от начала расчетного года.

Затраты и результаты, осуществляемые и получаемые до начала расчетного года, умножаются на α_t , а после начала расчетного года делятся на этот коэффициент.

Контрольные вопросы.

1. Факторы, вызывающие экономический эффект
2. Составляющие экономического эффекта
3. Хозрасчётные показатели экономической эффективности АСУТП
4. Расчет годового объема реализуемой продукции
5. Расчет изменения себестоимости продукции предприятия
6. Процедура расчета единовременных затрат на создание и внедрение АСУП

Тема 6. Методика определения трудоемкости разработки программ.

Метод оценки затрат труда основанный на опытно-статистических данных. Составляющие затрат труда при программировании. Качественные факторы и количественные коэффициенты увеличения затрат при программировании.

Метод оценки затрат труда основанный на опытно-статистических данных, полученных при подготовке задач к решению на компьютере.

В общем случае затраты труда, чел.-ч., определяются по формуле :

$$T = t_0 + t_i + t_a + t_n + t_{ot} + t_d, \quad (1)$$

где $t_0, t_i, t_a, t_n, t_{ot}, t_d$ - затраты труда соответственно на: подготовку описания задачи, на изучение описания задачи, разработку алгоритма решения задачи и составление проектных спецификаций

программы, программирование, отладку программы, подготовку документации по задаче (отчет, инструкции и т.д.).

Составляющие затрат труда, в свою очередь, можно определить через условное число команд в разрабатываемом программном обеспечении. В их число входят те команды, которые необходимо написать программисту в процессе работы над задачей с учетом возможных уточнений и совершенствования программы. Условное количество команд Q в программе задачи может быть определено по формуле:

$$Q = qC(1 + \sum_n P_{и}) \quad (2)$$

где q - предполагаемое число команд; C - коэффициент сложности программы; $P_{и}$ - коэффициент коррекции программы при ее разработке; n - количество коррекций программы в ходе ее разработки. Помимо названных выше используются и другие коэффициенты: квалификации разработчика K , затрат на алгоритмизацию A , увеличение затрат труда вследствие недостаточного или некачественного описания задачи B .

Используемые при оценке затрат труда на подготовку задач к решению на компьютере коэффициенты характеризуют различные факторы:

а) Коэффициент сложности программы C - относительную сложность программ задач по отношению к так называемой типовой задаче, сложность которой принята равной 1. За типовую задачу принимаются задачи учета, результат решения которых не выдается на печать, а после обновления храниться в памяти компьютера. Для задач оперативного управления коэффициент сложности программы равен - 1,25 - 1,5; планирование - 1,25 - 2,0.

б) Коэффициент коррекции программы $P_{и}$ - увеличение объема работ за счет внесения изменений в программу по результатам уточнения постановок и описания задач, изменения состава и структуры информации, а также уточнений, вносимых разработчиком для улучшения качества самой программы без изменения постановки задачи. На практике при разработке программы в среднем вносится 3-5 коррекций. Каждая из них ведет к переработке от 5 до 10% готовой программы.

в) Коэффициент квалификации разработчика K - степень подготовленности исполнителя к порученной ему работе. Этот коэффициент определяется в зависимости от стажа работы и составляет: для работающих до двух лет - 0,8; от 2 до 3 лет - 1; от 3 до 5 лет - 1,2.

г) Коэффициент затрат на алгоритмизацию A - трудоемкость разработки алгоритма по отношению к программированию. Этот коэффициент в общем случае является функцией сложности разрабатываемой программы и в силу его большой субъективности нами учитываться не будет.

д) Коэффициент увеличения затрат труда вследствие недостаточного описания задачи B - качество постановки задачи, выданной для разработки программы. В связи с тем, что задачи всегда в той или иной степени требуют уточнения и соответствующей доработки, $B \geq 1$. Практика показывает, что в большинстве случаев этот коэффициент в зависимости от сложности задач лежит в пределах от 1,2 до 1,5.

Затраты труда на подготовку описания задачи точно определить пока не предоставляется возможным, так как это связано с творческим характером работы. Практика показывает, что на разработку описания учетно-плановой задачи средней трудности требуется в зависимости от назначения и сложности задачи в среднем от 50 до 300 чел.-ч труда специалиста соответствующего профиля при хорошей организации работы у заказчика.

Затраты труда на изучение описания задачи $t_{и}$, чел.-ч., с учетом уточнения описания могут быть определены по формуле:

$$t_{и} = \frac{Q}{(75 \div 85)K} B \quad (3)$$

Затраты труда на разработку алгоритма задачи t_a , чел.-ч., рассчитываются по формуле:

$$t_a = \frac{Q}{(20 \div 25)K} \quad (4)$$

Затраты труда на составление программы по готовой блок-схеме при использовании алгоритмического языка $t_{п}$, чел.-ч.:

$$t_{п} = \frac{Q}{(20 \div 25)K} \quad (5)$$

Затраты труда на отладку программы tot, чел.-ч., определяются по формуле:

$$tot = \frac{Q}{(4 \div 5)K} \quad (6)$$

Затраты труда на подготовку документации по задаче тд, чел.-ч., рассчитываются по формуле:

$$td = t_{др} + t_{до}, \quad (7)$$

где $t_{др}$ - затраты труда на подготовку материала в рукописи, равные $Q/(15 \div 20)K$; $t_{до}$ - затраты труда на редактирование, печать и оформление документации, равные $0,75t_{др}$.

Контрольные вопросы.

1. Базовые факторы затрат труда при разработке программ.
2. Формула расчета затрат на основе статистического показателя количества команд.
3. Коэффициенты корреляции затрат.
4. Формулы уточнения затрат труда.

Тема 7. Надежность и качество функционирования ИС.

Определение понятий качественных характеристик, определение «надежности» технического объекта, свойства и стороны надежности. Виды надежности. Понятие отказов и их виды. Эффективность объекта и связь с надежностью.

Надежность — свойство объекта сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных режимах и условиях применения, технического обслуживания, ремонтов, хранения и транспортирования.

В этом определении на первый план ставится сохранение параметров, обеспечивающих выполнение заданных функций, что более точно выражает существо понятия «надежность».

Надежность является одной из важнейших характеристик качества объекта — совокупности свойств, определяющих пригодность использования его по назначению. В отличие от других характеристик качества надежность обладает следующей специфической особенностью. Обычные характеристики качества объекта, такие, как быстродействие, производительность, емкость памяти, мощность потребления, масса и др., измеряются для некоторого момента времени («точечные» характеристики качества). Надежность характеризует зависимость «точечных» характеристик качества либо от времени использования, либо от наработки объекта. Надежность — характеристика временная. Она может быть ориентирована либо на прошедшее время (в этом случае говорят: изделия до данного момента проработали такое-то количество часов, поэтому они обладали таким-то показателем надежности), либо в будущее время (в этом случае говорят: данные изделия, если они будут использоваться в предписанных условиях, будут обладать такой-то надежностью).

Надежность нельзя смешивать с другими характеристиками качества, игнорируя ее особенности, и противопоставлять им, так как в результате противопоставления возникают нелепые формулировки, вроде такой: «изделие высокого качества, но низкой надежности». Вывод о качестве объекта может быть сделан только тогда, когда учитываются и «точечные» характеристики качества и сохранение их в течение заданного интервала времени, или заданной наработки. Нельзя, например, говорить о качестве вычислительного комплекса по информации о его быстродействии и емкости памяти, если ничего не известно о его надежности. Точно так же нельзя говорить о качестве комплекса, если известно только то, что он обладает высокой надежностью.

Надежность — сложное свойство. Оно включает в себя более простые свойства (одно или несколько частных свойств). Эти частные свойства объекта называют также *сторонами надежности* (составными частями надежности).

Стороны надежности. К частным свойствам объекта, являющимся отдельными сторонами его надежности, относятся:

1. *Безотказность* — свойство объекта непрерывно сохранять работоспособность в течение некоторого времени, или некоторой наработки.

2. *Ремонтопригодность* — свойство объекта, заключающееся в приспособленности его к предупреждению и обнаружению отказов и восстановлению работоспособности объекта либо путем проведения ремонта, либо путем замены отказавших комплектующих элементов.

Поэтому возникают две самостоятельные характеристики ремонтнопригодности: *приспособленность к проведению ремонта* (ремонтопригодность в узком смысле) и *приспособленность к замене в процессе эксплуатации* (восстанавливаемость или заменяемость).

3. *Долговечность* — свойство объекта сохранять работоспособность до наступления предельного состояния, т. е. наступления такого состояния, когда оно должно быть направлено либо в ремонт (средний или капитальный), либо изъято из эксплуатации.

4. *Сохраняемость* — свойство объекта сохранять работоспособность в течение (и после) его хранения и (или) транспортирования. В данном случае частное свойство объекта — сохраняемость — также расчленяется на еще более простые свойства: сохраняемость в процессе (и после) хранения в специально приспособленном помещении, сохраняемость в процессе (и после) хранения в полевых условиях, сохраняемость в процессе (и после) транспортирования по железной дороге и т. д.

Работоспособность — такое состояние объекта, при котором он способен выполнять заданные функции, удовлетворяя требованиям нормативно-технической документации. Работоспособность — характеристика состояния объекта в некоторый момент времени.

Надежность — свойство сохранять работоспособность на некотором отрезке времени или при выполнении некоторого объема работы.

Таким образом, надежность — сложное комплексное свойство объекта — может расчленяться на все более простые (частные) свойства.

Перечисленные выше свойства объектов — частные свойства (стороны надежности) — являются общепризнанными и рекомендуются для широкого класса изделий. Однако для ИС, информационных сетей и вычислительной техники оказалось, что этих свойств характеристики надежности недостаточно. В практике создания и использования ИС находят применение дополнительные частные свойства, без учета которых нельзя в полной мере представить комплексное понятие «надежность». Рассмотрим эти свойства.

1. *Живучесть* — свойство объекта сохранять работоспособность (полностью или частично) в условиях неблагоприятных воздействий, не предусмотренных нормальными условиями эксплуатации.

При задании требований к надежности объекта обычно указываются нормальные условия его эксплуатации. Но к ряду объектов ответственного назначения могут предъявляться требования выполнить некоторые функции в условиях, существенно отличающихся от нормальных (даже катастрофически разрушающих). В этом случае возникает требование к живучести объекта. Оно может быть сформулировано, например, таким образом: «выполнять заданные функции на заданном интервале времени после разрушающего воздействия», «сохранять частичную работоспособность после разрушающего воздействия» и т. д. Главный смысл требования к живучести объекта состоит не только в том, чтобы он длительное время работал непрерывно без отказа в нормальных условиях эксплуатации его можно было быстро отремонтировать, но также и в том, чтобы он в ненормальных условиях эксплуатации сохранял работоспособность, хотя бы и ограниченную.

Достоверность информации, выдаваемой объектом. При работе вычислительной машины или тракта передачи информации могут отсутствовать отказы. Поэтому объект может обладать высокой безотказностью, хорошей долговечностью, сохраняемостью и ремонтпригодностью. Однако в нем могут иметь место сбои, искажающие информацию. В изделии «ломается», «портится» не аппаратура, а информация. Это не менее опасная «поломка», но она не находит отражения в перечисленных выше основных сторонах надежности. Поэтому и вводится еще одна дополнительная сторона надежности — достоверность.

В практике создания и использования технических систем возможно появление и других дополнительных сторон надежности. При введении в техническую документацию (ТЗ, ТУ) тех или других сторон надежности необходимо обращать внимание на то, чтобы среди них не было лишних, ненужных для описания комплексного показателя надежности, а также таких, которые нельзя измерить, которые не имеют определенного ясного и понятного физического смысла. При этом необходимо следить, чтобы набор используемых сторон надежности (частных свойств) был достаточным для описания надежности объекта, с тем чтобы объект, признанный удовлетворяющим требованиям надежности, не оказался вдруг непригодным для выполнения заданных функций.

Виды надежности. При исследовании надежности часто ставится задача определить причины, приводящие к формированию той или другой стороны надежности. Без этого невозможно наметить правильную программу работ по повышению надежности. Это приводит к делению надежности на:.'

- *аппаратурную надежность*, обусловленную состоянием аппаратуры;
- *программную надежность* объекта, обусловленную состоянием программ;
- *надежность объекта*, обусловленную качеством обслуживания, и *надежность функциональную*.

Аппаратурная надежность может при необходимости расчленяться на более мелкие разновидности надежности, например на *надежность конструктивно-схемную* и *производственно-технологическую*.

Особого внимания заслуживает понятие «программная надежность», так как ее важная роль в обеспечении надежности ИС является одной из самых характерных особенностей прикладной теории надежности ИС.

Учет влияния программного обеспечения приводит к необходимости выделять в особый вид программную надежность объектов.

Надежность функциональная — надежность выполнения отдельных функций, возлагаемых на систему. ИС, как правило, система многофункциональная, т. е. она предназначена для выполнения ряда функций, различных по своей значимости. Требования к надежности выполнения различных функций могут быть различным (например, для функции «расчет зарплаты» требуется высокая точность, но не требуется жесткого ограничения времени). Поэтому может оказаться целесообразным задавать различные требования к выполнению различных функций. При определении функциональной надежности следует учитывать всю совокупность факторов, действующих на показатель данной функциональной надежности. Примером функциональной надежности в ИС может быть надежность передачи определенной информации в системе передачи данных.

Большое число различных сторон и видов надежности не означает, что каждый раз при задании требований к надежности объектов или при исследовании его надежности необходимо использовать весь этот перечень. В каждом конкретном случае следует пользоваться такими сторонами и видами надежности, которые необходимы для характеристики надежности объекта с учетом его целевого назначения.

В основе понятия надежности объекта лежит понятие его отказа.

Отказы. *Отказ* объекта — событие, заключающееся в том, что объект либо полностью, либо частично теряет свойство работоспособности. При полной потере работоспособности возникает *полный отказ*, при частичной — *частичный отказ*. Понятия полного и частичного отказов каждый раз должны быть четко сформулированы перед анализом надежности, поскольку от этого зависит количественная оценка надежности. Требования к надежности изделия, а также количественная оценка надежности без указания признаков отказа не имеют смысла.

Отказы могут быть внезапными и постепенными. Эти отказы различны по природе возникновения.

Внезапному отказу может не предшествовать постепенное накопление повреждений, и он возникает внезапно. Технология изготовления современных элементов аппаратуры столь сложна, что не всегда удастся проследить за скрытыми дефектами производства, которые должны выявляться на стадии тренировки и приработки аппаратуры. В процессе эксплуатации случайно могут создаться условия, при которых скрытый дефект приводит к отказу изделия (пиковые нагрузки, тряска и вибрация, температурный скачок, помехи и т. д.). Но неблагоприятного сочетания неблагоприятных факторов может и не быть, тогда не будет и внезапного отказа. При большом уровне случайных неблагоприятных воздействий внезапный отказ может произойти даже при отсутствии скрытых дефектов.

Постепенный отказ возникает в результате постепенного накопления повреждений, главным образом вследствие износа и старения материалов.

Выделять внезапные и постепенные отказы необходимо, потому что закономерности, которым они подчиняются, различны. Различными поэтому должны быть и способы борьбы с этими отказами. Для уменьшения числа внезапных отказов может быть рекомендована предварительная тренировка и приработка изделия с целью выявления скрытых дефектов производства, а также ведение защиты от неблагоприятных воздействий типа помех, нагрузок, вибраций и т. п. Уменьшению числа постепенных отказов может содействовать своевременная замена сменных блоков, выработавших технический ресурс.

Отказ может быть кратковременным самоустраняющимся. В этом случае он называется *сбоем*. Характерный признак сбоя — то, что восстановление работоспособности после его возникновения не требует ремонта аппаратуры. Причиной сбоя может быть кратковременный отказ аппаратуры (например, залипание контакта), либо кратковременно действующая помеха, либо дефекты программы, приводящие к неблагоприятным временным характеристикам работы аппаратуры. Опасность сбоев заключается в том, что их трудно и часто даже невозможно обнаружить в процессе работы аппаратуры, но они могут исказить информацию настолько, что приведут к отказу выполнения заданной функции.

Отказы в ИС целесообразно подразделять на аппаратурные и программные.

Аппаратурным отказом принято считать событие, при котором изделие утрачивает работоспособность и для его восстановления требуется проведение ремонта аппаратуры или замена отказавшего изделия на исправное.

Программным отказом считается событие, при котором объект утрачивает работоспособность по причине несовершенства программы (несовершенство алгоритма решения задачи, отсутствие программной защиты от сбоев, отсутствие программного контроля состоянием изделия, ошибки в представлении программы на физическом носителе и т. д.). Характерным признаком программного отказа является то, что устраняется он путем исправления программ.

Для объектов ответственного назначения целесообразно выделять в отдельную группу отказы, которые могут приводить к катастрофическим последствиям (гибели людей и т. д.). В заданиях надежности необходимо выделять в отдельную группу требований по обеспечению безопасности.

В непосредственной связи с понятием «надежность» находит понятие «эффективность». *Эффективностью объекта* называется свойство объекта выдавать некоторый полезный результат (эффект) при использовании его по назначению.

Из определения надежности и эффективности очевидно, что это различные, хотя и взаимосвязанные, понятия. Чем выше надежность объекта, тем выше и его эффективность, но до некоторого предела. Повышение надежности выше определенного уровня нецелесообразно с точки зрения повышения эффективности.

Эффективность номинальная — эффективность объекта при безотказном его состоянии.

Эффективность реальная — эффективность реального объекта, т. е. не обладающего идеальной надежностью.

Эффективность техническая — технический эффект, полученный при использовании объекта (количество переданной информации, снижение затрат времени и т. д.).

Эффективность экономическая — степень выгодности экономических затрат при использовании объекта.

Эффективность оперативная — воздействие результатов применения объекта, на выполнение некоторой операции.

Например, снижение затрат времени на прибытие машины скорой помощи к пострадавшему — показатель технической эффективности ИС «Скорая помощь». Повышение числа благоприятных исходов при оказании медицинской помощи — показатель оперативной, эффективности ИС «Скорая помощь».

Использование понятия «эффективность» в теории надежности вызвано тем, что оно позволяет расширить представление о надежности сложных объектов, для которых возможны не только полные, но и частичные отказы. Используемые на первых этапах развития теории надежности показатели надежности были ориентированы на оценку надежности объектов, которые могут находиться только в двух состояниях — либо в состоянии отказа, либо в состоянии отсутствия отказа. Сложный объект может находиться в состоянии частичного отказа, т. е. в состоянии сохранения частичной работоспособности при отказе части комплектующих элементов. В этом случае необходимо определять степень значимости отказов — их влияние на эффективность объекта. В результате осуществляется слияние показателей надежности и показателей эффективности в комплексный показатель надежности, учитывающий влияние надежности на эффективность

Таким комплексным показателем надежности является коэффициент сохранения эффективности.

Тема 8. Управление коллективом

Социальная природа проблемных проектов. Меры эффективного управления людьми в интеллектуальной сфере. Интенсификация и качество работы. Ошибочность руководителя проекта. Внешняя среда и рациональное рабочее пространство. Факторы производительности. Закон Гильбо.

Социальная природа проблемных проектов

Около 15-25% крупных проектов заканчиваются «провалом» - отменяются, прерываются, откладываются. В большинстве случаев участники проектов причиной краха называют «политику», а не технику, технологии или экономику. Но трудности действительно политического толка составляют незначительную часть проблем.

Крупные проекты заканчиваются неудачей не из-за проблем с технологической и «политической» организацией производства, а в первую очередь из-за социологической природы коллектива

разработчиков. Человеческие взаимоотношения сложны, их проявления не бывают очевидными и прозрачными, но они имеют большее значение, чем любой другой аспект работы. Технические вопросы, вообще, решать проще, так как они очевиднее. Социология проекта и команды может выходить за рамки вашей компетенции, но не способностей.

Многие руководители сталкиваются больше с человеческим фактором, нежели с техническими сложностями. Для руководителей проектов рабочими единицами являются команды, проекты, инженеры, математики, техники, программисты и пр. и их координация – человеческое взаимодействие.

Как ни называй проблемы, связанные с людьми, именно они, вероятнее всего, станут причиной неприятностей в вашем следующем проекте, а вовсе не вопросы проектирования, реализации и методологии.

Ответ, возможно, кроется в явлении, которое мы окрестили миражами высоких технологий. Это распространенная убежденность людей, имеющих дело с любым аспектом новой технологии (а кто из нас не имеет?), что они работают в сфере действительно высоких технологий. Они потворствуют развитию этой иллюзии, объясняя на какой-нибудь вечеринке, что работают «в области компьютеров» или же «в области телекоммуникаций», намекая тем самым, что принадлежат к миру высоких технологий. Между нами говоря, обычно это не так. Исследователи, совершающие фундаментальные прорывы в перечисленных областях, действительно из мира высоких технологий. А мы лишь используем результаты их труда. Компьютеры и прочие элементы новых технологий служат нам для организации собственных предприятий. Наш успех напрямую зависит от качественного человеческого взаимодействия всех участников предприятия, а наши неудачи являются прямым следствием недостатка человеческого взаимодействия.

Проблемы в работе коллектива разработчиков имеют не столько технологическую, сколько социальную природу

Основные меры эффективного управления людьми в интеллектуальной сфере.

1. Итеративное проектирование. Допустимость ошибок.

Для большинства работников, занятых в сфере интеллектуального труда, допускаемые время от времени ошибки - вполне естественная и безопасная составляющая деятельности. Но иногда между ошибкой в работе и грехом проводятся почти библейские ассоциации. Для того чтобы изменить подобное отношение, необходимо принять действительно серьезные меры.

Допустимость ошибок является составной частью парадигмы *итеративного проектирования*. Идея заключается в том, что некоторые проектные решения изначально содержат недостатки, и такие решения следует выбрасывать, а не пытаться исправить. При проектировании следует ожидать появления подобных тупиков. Усилия, потраченные на тупиковую разработку, — небольшая цена за возможность начать с чистого листа. К нашему удивлению, многие руководители считают, что такой подход станет неразрешимой политической проблемой для их собственного начальства: «Как мы можем выбросить продукт, в производство которого наша компания вложила деньги?» Похоже, они считают, что лучше все же спасти дефектный вариант, пусть это в перспективе и обойдется дороже.

Поощрение атмосферы, не позволяющей допускать ошибки, просто заставляет людей занимать оборонительные позиции. Они не пробуют подходы, способные закончиться отрицательным результатом. Вы же поощряете такое поведение, пытаясь упорядочить процесс и использовать жесткие методологии, ради исключения ошибок запрещающие сотрудникам принимать ключевые стратегические решения. Любые меры, препятствующие совершению ошибок, могут лишь ненадолго поднять уровень технологии, а вот социология команды может пострадать весьма серьезно.

Противоположный подход - *поощрение* нечастых ошибок. Время от времени спрашивайте у своих подчиненных, с какими тупиковыми ситуациями они столкнулись, и постарайтесь объяснить, что ответ «такого не было» - не самый лучший. Когда люди совершают ошибки, их следует поздравлять, потому что они получают деньги в том числе и за ошибки.

Необходимо помнить, что крупные проекты строятся на стратегии итеративного или спирального проектирования. Эти стратегии используют подход «поощрения нечастых ошибок». Общайтесь с людьми, выясняйте их тупики и ошибки, не ругайте, а наоборот поддерживайте - «Только дурак никогда не ошибается», «За одного битого двух небитых дают».

2. Принцип «кнута и пряника». Самое печальное в таком подходе – это то, что он практически никогда не оправдан, если применяют его в грубой форме наказаний и поощрений. Особенно это характерно для людей с творческим потенциалом – они и так любят свою работу, и подгонять их не надо. Это может вызвать только негативную реакцию. Тоже и с Вашими «пряниками» - ну не любит сладкое именно этот разработчик.

Даже если «педалирование» дает кратковременный прирост производительности, в перспективе оно может оказать отрицательный эффект: нет ничего более обескураживающего для работника, чем чувство, что его собственной мотивации не хватает и требуется «добавка» руководства

3. Постоянное развитие и обновление проектов.

Стабилизация проекта означает его смерть. Образ мышления, характерный для производства, находящегося в стабильном состоянии, особенно плохо сочетается с ведением проектов. Мы склонны забывать, что самоцель существования проекта состоит в том, чтобы создать что-то новое, так или иначе – совершенствовать бизнес. Единственное *стабильное* состояние в жизни проекта - трупное окоченение. Управление проектом должно сосредоточиваться на *динамике* его развития, разумеется, за исключением случаев, когда вы управляете проектом уже прерванным или находящимся на грани отмены. Но почему-то ценность людей для нового проекта мы вычисляем, исходя из их стабильных показателей: сколько кода они могут написать или сколько могут задокументировать. И слишком мало внимания уделяем тому, какой именно качественный *вклад* каждый из них может внести в развитие предприятия.

4. Человеческие запчасти

В производственной сфере людей удобно считать частями рабочего механизма. Когда часть механизма изнашивается, ее заменяют на другую. Замена совместима с оригиналом, поэтому новую можно, грубо говоря, заказать по каталогу.

Многие руководители проектов по разработке придерживаются такого же подхода. Они идут на все, убеждая себя, что нет незаменимых людей. Из страха, что ключевой человек уйдет, они заставляют себя считать, что ключевых людей попросту не существует. Разве не в этом, в конце концов, сущность менеджмента - делать так, чтобы работа не останавливалась в случае ухода отдельных людей?

Интенсификация

Увеличение производительности по идее означает увеличение объема полезной работы, выполняемой за определенный период времени, но слишком часто на практике выходит, что это стремление получить больше за оплаченный час работы. Разница велика. Руководители мечтают о достижении новых уровней производительности посредством простого механизма неоплаченных сверхурочных. Они делят выполненную в течение недели работу на сорок часов, а не на те восемьдесят или девяносто, которые в действительности затратил работник.

Это не совсем производительность (скорее мошенничество), но такова повседневная деятельность многих американских менеджеров. Кнутом и пряником они заставляют своих людей работать дольше. Они заставляют людей принимать безнадежно жесткие графики, заставляют принести в жертву все, чтобы уложиться в сроки, и делают все возможное, чтобы заставить людей работать дольше и усерднее.

Ваши люди очень остро понимают, что человеку дана лишь одна короткая жизнь, и они хорошо знают, что в этой жизни должно быть что-то поважнее, чем их глупая работа.

Сверхурочные сотрудников, сидящих на окладах, - это плод воображения наивного руководителя. За сиюминутной очевидной пользой, всегда следует равносильные компенсации. В конечном итоге каждый час сверхурочных компенсируется часом недоработки. В краткосрочной перспективе такой компромисс может принести пользу проекту, но в долгосрочной польза нивелируется.

Трудоголики не компенсируют свои сверхурочные. Они согласны работать лишние часы, пусть и с пониженной эффективностью. Но это все лишь на время. Осознание того, что большие ценности жизни принесены в жертву меньшим, разрушительно. Он просто уйдет – еще один человек сгорел на работе.

«Под давлением люди работают не лучше, а всего лишь быстрее».

Качество

Руководители, склонны считать качество лишь одним из свойств продукта, чем-то, что можно регулировать в зависимости от потребностей рынка. Качество - вроде клубничного сиропа на пломбире - каждый добавляет по вкусу.

Взгляд создателя на качество совсем иной. Поскольку самооценка сил связана с качеством продукта, он, создатель (разработчик), устанавливает собственные стандарты. Удовлетворительный для него минимум более-менее соответствует высшему уровню качества, достигнутому им в прошлом. И это неизбежно более высокий стандарт, чем тот, что необходим рынку и за который рынок готов платить.

Мы должны исходить из того, что люди, финансирующие наш труд, находятся в здравом уме и способны принять разумный компромисс между качеством и стоимостью. Основная идея здесь в том, что осознаваемая клиентом потребность в качестве зачастую не столь велика, как аналогичная потребность разработчика. Существует естественное противоречие. Снижение качества продукта, вероятно, приведет к тому, что некоторые люди откажутся от покупки, но сокращение рыночной доли - прямое следствие любого подобного снижения качества - зачастую компенсируется увеличением прибылей на каждую проданную единицу продукта.

Бегство от совершенства - так мы называем ситуацию, когда стандарты качества устанавливает покупатель, а не создатель. Стандарты, основанные на существующем рыночном качестве, имеют смысл лишь до тех пор, пока вы не обращаете внимания на настроение и эффективность труда разработчиков.

В долгосрочной перспективе рыночные стандарты качества обходятся дороже. Какой урок отсюда следует?

Качество, серьезно превосходящее запросы конечного пользователя, есть средство достижения более высокой производительности.

На самом деле идею связи качества и производительности можно трактовать иными словами:

Качество не стоит ничего, но только для тех, кто готов дорого за него заплатить.

В здоровой рабочей обстановке причинами стагнации для некоторых людей становятся недостаток компетенции, нехватка уверенности, а также неопределенность цели проекта и отсутствие интеграции с командой. Ни в одном из этих случаев временное давление помочь не способно. Скажем, когда сотрудник вяло работает и кажется, что он не задумывается о качестве своих результатов, это верный признак того, что бедняга подавлен сложностью работы. Ему не нужно более сильное давление. Ему нужна смена деятельности, возможно, просто перевод в другую компанию.

Плохие прогнозы всегда являются фактором, снижающим мотивацию. Системные аналитики более склонны к точным оценкам, чем программисты или руководители. Системный аналитик, как правило, владеет той же информацией о поставленной задаче, но его не сдерживает природный оптимизм исполнителя или политические и финансовые взгляды шефа. Более того, системные аналитики имеют больший опыт в оценке, они способны предсказывать результаты более точно, поскольку уже занимались этим в прошлом и делали соответствующие выводы.

Ложные надежды, порождаемые простыми технологическими способами нерешения проблемы, подобны Сиренам, соблазнявшим несчастного Одиссея.

Семь ложных надежд руководителя проекта по разработке программного обеспечения

1. Существует еще не изученный новый трюк, который поднимет производительность до небес.

Ответ: вы недостаточно глупы, чтобы пропустить нечто столь фундаментальное. Вы постоянно исследуете новые подходы и пробуете те, что кажутся наиболее осмысленными. Ни одна из мер уже принятых и ни одна из тех, что вы еще можете принять, не позволит поднять производительность до небес. Что они, однако, могут, - так это держать всех в форме: людям нравится думать, учиться и расти. Мысль же о том, что существует волшебное новшество, которое пропустили, - это мысль пораженческая, и ею умело пользуются те, кто волшебные новшества продают.

2. Другие руководители умудряются получить прирост производительности в сто, двести и даже более процентов!

Ответ: Забудьте об этом. Как правило, волшебный инструмент, который вам предлагают, работает на этапах программирования и тестирования. Но как только программирование и тестирование, заканчиваются, уже невозможно получить прирост в сто процента! Остаются еще анализ, контракты, спецификации, подготовка, приемка, реконструкция и переключение с одной системы на другую.

3. Технологии развиваются с такой скоростью, что невозможно за всем успевать.

Ответ: Да, технологии развиваются быстро, но (вот опять иллюзия высоких технологий) ваши труды по большей части к высоким технологиям не относятся. Машины изменились очень сильно, но бизнес, связанный с разработкой программного обеспечения, всегда был довольно статичен. Мы все так же тратим большую часть времени на требования и спецификации, на низкотехнологичные аспекты нашей работы. Производительность индустрии программного обеспечения растет на 3-5% в год, что лишь немногим больше, чем показатели в автомобильной или сталелитейной индустрии.

4. Смена языка даст гигантские преимущества.

Ответ: Выбор языка имеет значение, потому что он влияет на способ решения проблемы, но опять же, язык оказывает влияние лишь на этапе реализации. Благодаря преувеличениям, некоторые из новых языков попадают в разряд плацебо. Если последние несколько десятилетий вы не проспали у голубого телевизора, то смена языка не сильно вам поможет.

5. Из-за отставания следует немедленно удвоить производительность.

Ответ: Отставание разработки, о котором так много говорят - миф. Все мы знаем, что в конечном итоге любой проект обходится дороже, чем было запланировано в начале. Поэтому стоимость системы, не созданной в прошлом году (потому что не было ресурсов) принято оптимистично считать равной половине стоимости этой же системы, будь она создана, или даже менее того. Проект, попавший в капкан мифического отставания, находится там потому, что выгода от него явно недостаточна, чтобы служить причиной этот проект затевать, даже если оценки стоимости крайне оптимистичны. Если бы вы знали реальную стоимость проекта, то верно понимали бы его смысл: экономическая неудача.

6. Все уже автоматизировано; не пора ли напроць автоматизировать персонал, разрабатывающий программное обеспечение?

Ответ: Это еще одна вариация на тему иллюзии высоких технологий - вера в то, что разработчики программ выполняют работу, легко поддающуюся автоматизации. Их основная работа - человеческое взаимодействие, позволяющее преобразовать изложенные пользователями потребности в формальное представление. Кто-то должен делать эту работу независимо от того, какие формы принимает цикл жизни продукта. И вряд ли возможно данную задачу автоматизировать.

7. Люди будут лучше работать, если как следует на них надавить.

Ответ: Нет, не будут. Но будут меньше любить свою работу

Как видим, один сплошной негатив. Если давление на людей способствует снижению производительности, а установка самых последит техно-фенечек тоже не сильно спасает, чем же должен заниматься руководитель?

«Назначение руководителя не в том, чтобы заставить людей работать, а в том, чтобы создать им условия для работы».

Внешняя среда и рациональное рабочее пространство.

Материал посвящен некоторым причинам потери времени и мерам, которые можно предпринять для создания стабильной, способствующей работе обстановки.

Для того чтобы дать людям возможность работать, необходимо справиться со всеми теми факторами, которые иногда делают работу невозможной. Причины потерь часов и дней многочисленны, но они не очень отличаются одна от другой. Чаще всего это в той или иной форме недостатки среды, в которой приходится работать. Телефон звонит не переставая, мастер по ремонту принтеров остановился поболтать, копировальная машина сломалась, какой-то гражданин звонит, чтобы уточнить время сдачи крови. Отдел кадров продолжает вопить, что нужны обновленные отчеты о квалификации сотрудников, расписание сдавать в три часа дня, снова телефонные звонки в безумных количествах... и день пропал. В некоторые дни невозможно потратить хотя бы минуту на что-либо, имеющее отношение к непосредственной задаче.

Если бы только руководители подвергались всем этим диверсиям, ситуация была бы не столь ужасной - сотрудники могли бы мирно работать. Но, как вам известно, на практике все не так. Рабочий день каждого человека буквально зачумлен неожиданными перерывами и крушениями планов. Пропадают целые дни, и никто не может четко объяснить, куда они подевались. Если интересно, почему все вокруг происходит вопреки вашим планам, подумайте вот о чем:

Есть миллион способов потерять рабочий день и ни одного, который бы компенсировал потерю.

Рациональное рабочее пространство.

1. Рабочий стола: поверхность, оборудование, электропроводка, полки, тумбочки и пр.
2. Защита от шума, света, вибрации и т.д.
3. Создание социальной зоны – места взаимного общения, релаксации и т.д.
4. Разрешение на индивидуальные украшения рабочего пространства – фотографии, сувениры, амулеты, безделушки и пр.



Рисунок 1. Масштаб различий между индивидуумами.

На рис.1 представлены результаты, полученные из различных источников, и он иллюстрирует масштабы различий между индивидуумами.

Похоже, что при измерении вариаций производительности для выборки индивидуумов действуют три основных правила:

- Отношение производительности лучших сотрудников к производительности худших составляет примерно 10:1.
- Наиболее производительный сотрудник в 2,5 раза более производителен, чем средний.
- Наиболее производительная половина сотрудников имеет в 2 раза большую производительность, чем менее производительная половина.

Эти правила действуют практически на любые параметры производительности, которые возможно определить. Так, к примеру, лучшая половина выборки сделает заданную работу минимум в полтора раза быстрее остальных; представители другой половины, которые чаще ошибаются, сделают две трети всех ошибок и так далее.

Среди обнаруженных факторов, положительно влияющих на производительность, оказался и весьма неожиданный: *большое значение - выбор напарника*. Если вам доставался производительный напарник, то получалось и у вас. Если Ваш напарник никак не мог закончить работу - не могли закончить ее и Вы.

Факторы производительности

Исследуя результаты состязаний, мы обнаружили, что следующие факторы слабо влияли на производительность или не влияли вовсе:

– *Язык*: Разработчики, использовавшие старые языки, выполняли задание не хуже тех, кто писал на более «продвинутых», но одного и того же уровня технологий (Fortran, PL, Pascal, C). Внутри каждой языковой группы распределение производительности было таким же, как в целом по выборке. Единственным исключением из этого наблюдения стал язык ассемблера: участники, писавшие на ассемблере, сильно отстали от всех остальных языковых групп. (Впрочем, люди, пишущие на ассемблере, привыкли к такому положению вещей)

– *Опыт*: Люди с десятилетним опытом не превосходили по производительности тех, у кого опыта было всего два года. Опыт и производительность никак не коррелировали, разве что люди, менее шести месяцев имевшие дело с языком, работали не так эффективно, как участники.

– *Количество недочетов*: Около трети участников выполнили упражнение с нулевым количеством недочетов. В целом не было отмечено снижения производительности из-за более высокой точности работы. Более того, в среднем эта треть участников выполняла задание быстрее чем участники, допускавшие недочеты.

– *Зарплата*: Уровень зарплаты достаточно сильно варьировался. Между зарплатой и производительностью наблюдалась очень слабая связь. Лучшая половина получала на десять процентов больше худшей, но работала почти в два раза эффективнее.

Тенденция становится очевидной, если проследить характеристики рабочих мест тех, кто хорошо показал себя в соревнованиях (по совокупности параметров производительности), и сравнить их результаты с результатами участников, не проявивших себя столь хорошо. Мы решили сравнить первую четверть участников с четвертой. Средняя производительность участников первой четверти в 2,6 раза выше средней производительности участников последней четверти. Свойства рабочей среды приведены табл.1;

Таблица 1. Свойства среды лучших и худших в плане производительности, участников военных маневров разработчиков

Свойство среды	Первая четверть	Последняя четверть ,
1. Какова площадь вашего рабочего места?	7 кв.м.	4,1 кв.м.
2. Достаточно ли тихо на рабочем месте?	57% ответили «Да»	29% ответили «Да»
3. Достаточно ли уединенное рабочее место?	62% ответили «Да»	19% ответили «Да»
4. Можете ли вы отключить свой телефон?	52% ответили «Да»	10% ответили «Да»
5. Можете ли вы перенаправлять свои звонки?	76% ответили «Да»	19% ответили «Да»
6. Часто ли люди прерывают вас без необходимости?	38% ответили «Да»	76% ответили «Да»

Первая четверть, то есть участники, быстрее и эффективнее других выполнившие задание, имеют рабочие места, существенно отличающиеся, от тех, что достались последней четверти. Наиболее производительно работают в более тихих, более уединенных, более просторных и лучше защищенных от внешнего вмешательства местах

Закон Гильбо

Если вам нужно представить что-либо в количественной форме, это можно измерить каким-то способом, который даст лучшие результаты, чем в случае, если не проводить измерений вовсе.

Закон Гилба не гарантирует, что измерения будут бесплатными или даже дешевыми, и они могут оказаться далекими от идеала - но это все равно лучше, чем ничего.'

Конечно же, производительность поддается измерению. Если создать группу из людей, выполняющих одинаковую или похожую работу, и дать им день на создание разумной схемы самоконтроля в целях измерения, они в результате создадут нечто, подтверждающее закон Гилба. Цифры, полученные этими людьми, позволят им регулировать собственную производительность, а также в сочетании с кружками качества или каким-либо другим способом взаимного контроля предоставят возможность учиться на методах работы коллег. Средние показатели, вычисленные для этой группы, дадут руководству надежное свидетельство воздействия на производительность таких событий, как улучшение офисной среды.

В наиболее знакомой нам области конструирования программного обеспечения существует неограниченное число работающих механизмов измерения производительности. Существует даже служба оценки производительности, которая может приехать и показать, где находится Ваша организация в сравнении с другими участниками отрасли. Организация, не способная выполнить какую-либо оценку собственной производительности в программировании, просто недостаточно прилежно старается это сделать.

Учитывая, что организации на практике довольно значительно отличаются одна от другой, Вы просто не можете себе позволить игнорировать информацию о собственном текущем положении. Ваши конкуренты могут работать в десять раз эффективнее. Если вы не знаете об этом, то не можете принять меры. Зато рынок все поймет. Он предпримет собственные шаги для выправления ситуации, и эти шаги будут не в Вашу сторону.

Тема 9. Модели и метрики оценки качества ПО

Факторы противоречивости применения формальных оценок. Классификация метрик. Список метрик.

[http://guap.ru/dept04/caf46/textbooks/std_pro/face.htm

Богданов Д.В., Путилов В.А., Фильчаков В.В. Стандартизация процессов обеспечения качества программного обеспечения. - Апатиты, КФ ПетрГУ, 1997. – 161]

Современная наука о разработке и оценки качества программного обеспечения имеет в своем арсенале внушительный инструментарий метрик и моделей, позволяющих проводить оценку различных как частных так и обобщенных производственных и эксплуатационных характеристик и свойств ПО.

Практика сегодняшнего дня программной индустрии демонстрирует недоверие разработчиков и пользователей ПО к существующим методикам, а также продолжающиеся научные споры о достоверности тех или иных мер. Отметим факторы вызывающие столь противоречивое отношение ученых к вроде бы сугубо формальным показателям:

- игнорирование парадигм системного анализа;

- слабая или отсутствующая корреляция целей проекта с предметной областью и областью применения ПО;
- не учитываются взаимосвязанные уровни сложности: сложность ПО – сложность технологического процесса – жизненный цикл;
- ффивольное, научно необоснованное использование метрик в разноплановых задачах приводит к противоречию с реальной измеряемой мерой.

Тем не менее, анализ технологического процессов производства ПО убедительно показывает, насколько дорого обходится несовершенство научного прогноза оценки сложности программ, контроля и управления их разработкой, выбираемого технологического инструментария, что и приводит в конечном итоге к несоответствию требованиям пользователя, стандартам и, как следствие, повышение трудозатрат, а порой и неудача всего проекта.

Эти и ряд других практических реалий являются следствием отсутствия сквозной методической поддержки, для реализации которой требуется:

- научно-обоснованный отбор методик, моделей, методов оценки;
- выбор ограничений их пригодности для различного класса задач;
- учет (выбор или комбинацию) методологических подходов (каскадная, итерационная, спиральная, объектно-ориентированная и пр.) для реализации технологических процессов разработки;
- применение адекватного моделирования для измерения соответствующих показателей;
- применения избыточного многовариантного модельного исследования показателей для повышения достоверности их оценок;
- накопления и интеграции разнородной метрической информации для принятия своевременных производственных решений.

Важнейшей целью научных исследований в технологиях разработки ПО является уменьшение ее сложности. Именно это позволяет снизить трудоемкость проектирования, изготовления, испытаний и сопровождения ПО, повысить эксплуатационные характеристики – надежность, производительность.

Целенаправленное снижение сложности ПО представляет собой многошаговую процедуру и требует предварительного исследования существующих показателей сложности, проведения их классификации и соотнесения с типами программ и их местоположением в жизненном цикле.

В таблице 1 представлены разнообразные метрики сложности ПО для различных форм их представления, хорошо зарекомендовавшие себя при оценке качества ПО, пригодные для прогнозирования и констатации различных показателей сложности и надежности.

Таблица 1 МЕТРИКИ СЛОЖНОСТИ

Название метрики	Формула, обозначение
<p>Метрики Холстеда</p> <ul style="list-style-type: none"> - длина программы; - объем программы - оценка ее реализации; - трудность ее понимания; - трудоемкость кодирования; - уровень языка выражения; - информационное содержание; - оптимальная модульность; 	$N = n_1 \log_2 n_1 + n_2 \log_2 n_2$ $V = N \log_2 n$ $L^* = (2 n_2) / (n_1 N_2)$ $E_c = V / L^*$ $D = (n_1 N_2) (2 n_2) = 1 / L^*$ $\lambda^* = V / D^2 = V / L^{*2}$ $I = V / D$ $M = n_2^* / 6$
<p>Метрики Джилба</p> <ul style="list-style-type: none"> - количество операторов цикла; - количество операторов условия; - число модулей или подсистем; 	L_{loop} L_{if} L_{mod}

<ul style="list-style-type: none"> - отношение числа связей между модулями к числу модулей; - отношение числа ненормальных выходов из множества операторов к общему числу операторов; 	$f = N_{SV}^4 / L_{mod}$ $f^* = N_{SV}^* / L$
<p>Метрики Мак-Кейба</p> <ul style="list-style-type: none"> - цикломатическое число; - цикломатическая сложность; 	$\lambda(G) = m - n + p$ $v(G) = \lambda(G) + 1 = m - n + 2$
<p>Метрика Чепена</p> <ul style="list-style-type: none"> - мера трудности понимания программ на основе входных и выходных данных; 	$H = 0.5T + P + 2M + 3C$
<p>Метрика Шнадевида</p> <ul style="list-style-type: none"> - число путей в управляющем графе 	$S = \sum P_i C_i$
<p>Метрика Майерса</p> <ul style="list-style-type: none"> - интервальная мера; 	$[v_1 \div v_2]$
<p>Метрика Хансена</p> <ul style="list-style-type: none"> - пара (цикломатическое число, число операторов) 	$\{v, N\}$
<p>Метрика Чена</p> <ul style="list-style-type: none"> - топологическая мера Чена; 	$M(G) = (v(G), N, Q_0)$
<p>Метрика Вудворда</p> <ul style="list-style-type: none"> - узловая мера (число узлов передач управления); 	$\Psi \xi$
<p>Метрика Кулика</p> <ul style="list-style-type: none"> - нормальное число (число простейших циклов в нормальной схеме программы); 	$Norm(P)$
<p>Метрика Хура</p> <ul style="list-style-type: none"> - цикломатическое число сети Петри, отражающей управляющую структуру программы; 	$\lambda(G_p^*)$
<p>Метрики Витворфа, Зулевского</p> <ul style="list-style-type: none"> -мера сложности потока управления -мера сложности потока данных; 	$\gamma(P)$ $\Omega(P)$
<p>Метрика Петерсона</p> <ul style="list-style-type: none"> - число многовходовых циклов; 	Nm_{loop}
<p>Метрики Харрисона, Мэйджела</p> <ul style="list-style-type: none"> - функциональное число (сумма приведенных сложностей всех вершин управляющего графа); - функциональное отношение (отношение числа вершин графа к функциональному числу); - регулярные выражения (число операндов, операторов и скобок в регулярном выражении управляющего графа программы); 	$f_1 = \sum \chi_1$ $f^* = N \chi_1 / f_1$ $p(G) = N + L + Sk$
<p>Метрика Пивоварского</p> <ul style="list-style-type: none"> - модифицированная цикломатическая мера сложности; 	$N(G) = n^*(G) + \sum P_i$
<p>Метрика Пратта</p>	

- тестирующая мера;	Test (Pr)
Метрика Кантоне - характеристические числа полиномов, описывающих управляющий граф программы;	PCN*
Метрика Мак-Клур - мера сложности, основанная на числе возможных путей выполнения программы, числе управляющих конструкций и переменных;	$C(V) = D(V) \times J(V) / N$
Метрика Кафура - мера на основе концепции информационных потоков;	I(G)
Метрика Схуттса, Моханти - энтропийные меры;	$\varepsilon (G)$
Метрика Коллофело - мера логической стабильности программ;	h (G)
Метрика Зольновского, Симмонса, Тейера Взвешенная сумма различных индикаторов: - (структура, взаимодействие, объем, данные); - (сложность интерфейса, вычислительная сложность, сложность ввода/вывода, читабельность);	$\sum (\alpha, \beta, \gamma, v)$ $\sum (\chi, X, v, \pi)$
Метрика Берлингера - информационная мера;	$I(R) = m (F^*(R) \times F(R))^2$
Метрика Шумана - сложность с позиции статистической теории языка;	$\Xi (\Psi)$
Метрика Янгера - логическая сложность с учетом истории вычислений; Сложность проектирования Насыщенность комментариями Число внешних обращений Число операторов	$\Lambda (\omega)$ $C_c = \sum \log_2 (i + 1) [\sum_n C_{xy} (n)]$ $X = K/C$ C_i L_1

Показателей сложности (в той или иной степени) велико и разнообразно. Как правило, с понятием сложность программ, связывают технологический аспект управления качеством ПО и контроль технической сложности в период эксплуатации. Проведем ряд классификаций.

Производственная классификация. Показатели сложности делятся на две группы: сложность проектирования и сложность функционирования.

1. Сложность проектирования, которая определяется размерами программы, количеством обрабатываемых переменных, трудоемкостью и длительностью разработки и др. факторами, анализируется на основе трех базовых компонентов: сложность структуры программы, сложность преобразований (алгоритмов), сложность данных.

2. Эксплуатационные качества, характеризующие группу показателей относящиеся к техническим характеристикам: временная, программная и информационная сложности.

Классификация по формальным признакам моделирования. Выделим основные формальные составляющие основу типизации метрик сложности, которые базируются на одинаковой формальной основе:

- словарная метрика, основанная на метрических соотношениях Холстеда, цикломатических мерах Мак-Кейба и измерениях Тейера.
- метрики связности ориентированы на связи, отражающих сложность отношений между компонентами системы - это метрики Уина и Винчестера.
- архитектурные метрики, связанные с понятием «структурного», «модульного» построения программ и их оформлением.
- метрики, основанные на идеях энтропии.

Существует еще ряд подходов к классификации мер сложности, однако они, фиксируя частные стороны исследуемых программ, не позволяют (пусть с большим допущением) отразить **общее**, то, чьи замеры могут лечь в основу производственных решений.

Первой топологической мерой сложности является цикломатическая мера Мак-Кейба. В ее основе лежит идея оценки сложности ПО по числу базисных путей в ее управляющем графе, т.е. таких путей, komponуя которые можно получить всевозможные пути из входа графа в выходы. Цикломатическое число $\lambda(G)$ орграфа G с n -вершинами, m -дугами и p -компонентами связности есть величина $\lambda(G) = m - n + p$.

Имеет место теорема о том, что число базисных путей в орграфе равно его цикломатическому числу, увеличенному на единицу. При этом, цикломатической сложностью ПО P с управляющим графом G называется величина $v(G) = \lambda(G) + 1 = m - n + 2$. Практически цикломатическая сложность ПО равна числу предикатов плюс единица, что позволяет вычислять ее без построения управляющего графа простым подсчетом предикатов. Данная мера отражает психологическую сложность ПО.

К достоинствам меры относят простоту ее вычисления и повторяемость результата, а также наглядность и содержательность интерпретации. В качестве недостатков можно отметить: нечувствительность к размеру ПО, нечувствительность к изменению структуры ПО, отсутствие корреляции со структурированностью ПО, отсутствие различия между конструкциями Развилка и Цикл, отсутствие чувствительности к вложенности циклов. Недостатки цикломатической меры привело к появлению ее модификаций, а также принципиально иных мер сложности.

Дж. Майерс предложил в качестве меры сложности интервал $[v_1 \div v_2]$, где v_1 - цикломатическая мера, а v_2 - число отдельных условий плюс единица. При этом, оператор DO считается за одно условие, а CASE с n - исходами за $n - 1$ - условий. Введенная мера получила название интервальной мерой.

У. Хансену принадлежит идея брать в качестве меры сложности ПО пару {цикломатической число, число операторов}. Известна топологическая мера $Z(G)$, чувствительная к структурированности ПО. При этом, она $Z(G) = V(G)$ (равна цикломатической сложности) для структурированных программ и $Z(G) > V(G)$ для неструктурированных. К вариантам цикломатической меры сложности относят также меру $M(G) = (V(G), C, Q)$, где C - количество условий, необходимых для покрытия управляющего графа минимальным числом маршрутов, а Q - степень связности структуры графа программы и ее протяженность.

К мерам сложности, учитывающим вложенность управляющих конструкций, относят тестирующую меру M и меру Харрисона-Мейджела, учитывающих уровень вложенности и протяженности ПО, меру Пивоварского - цикломатическую сложность и глубину вложенности, и меру Мак-Клура - сложность схемы разбиения ПО на модули с учетом вложенности модулей и их внутренней сложности.

Функциональная мера сложности Харрисона-Мейджела предусматривает приписывание каждой вершине графа своей собственной сложности (первичной) и разбиение графа на сферы влияния предикатных вершин. Сложность сферы называют приведенной и слагают ее из первичных сложностей вершин, входящих в сферу ее влияния, плюс первичную сложность самой предикатной вершины. Первичные сложности вычисляются всеми возможными способами. Отсюда функциональная мера сложности ПО есть сумма приведенных сложностей всех вершин управляющего графа.

Мера Пивоварского ставит целью учесть в оценке сложности ПО различия не только между последовательными и вложенными управляющими конструкциями, но и между структурированными и неструктурированными программами. Она выражается отношением $N(G) = v^*(G) + \sum P_i$, где $v^*(G)$ -

модифицированная цикломатическая сложность, вычисленная так же, как и $V(G)$, но с одним отличием: оператор CASE с n - выходами рассматривается как один логический оператор, а не как $n - 1$ операторов. P_i - глубина вложенности i -той предикатной вершины.

Для подсчета глубины вложенности предикатных вершин используется число сфер влияния. Под глубиной вложенности понимается число всех сфер влияния предикатов, которые либо полностью содержатся в сфере рассматриваемой вершины, либо пересекаются с ней. Глубина вложенности увеличивается за счет вложенности не самих предикатов, а сфер влияния. Сравнительный анализ цикломатических и функциональных мер с обсуждаемой для десятка различных управляющих графов программы показывает, что при нечувствительности прочих мер этого класса, мера Пивоварского возрастает при переходе от последовательных программ к вложенным и далее к неструктурированным.

Мера Мак-Клура предназначена для управления сложностью структурированных программ в процессе проектирования. Она применяется к иерархическим схемам разбиения программ на модули, что позволяет выбрать схему разбиения с меньшей сложностью задолго до написания программы. Метрикой выступает зависимость сложности программы от числа возможных путей исполнения, числа управляющих конструкций и числа переменных (от которых зависит выбор пути). Методика расчета сложности по Мак-Клуру четко ориентирована на хорошо структурированные программы.

Тестирующей мерой M называется мера сложности, удовлетворяющая следующим условиям:

1. Мера сложности простого оператора равна 1;
2. $M(\{F_1; F_2; \dots; F_n\}) = e^n M(F_i)$;
3. $M(\text{IF } P \text{ THEN } F_1 \text{ ELSE } F_2) = 2 \text{ MAX}(M(F_1), M(F_2))$;
4. $M(\text{WHILE } P \text{ DO } F) = 2 M(F)$.

Мера возрастает с глубиной вложенности и учитывает протяженность программы. К тестирующей мере близко примыкает мера на основе регулярных вложений. Идея этой меры сложности программ состоит в подсчете суммарного числа символов (операндов, операторов, скобок) в регулярном выражении с минимально необходимым числом скобок, описывающим управляющий граф программы. Все меры этой группы чувствительны к вложенности управляющих конструкций и к протяженности программы. Однако возрастает уровень трудоемкости вычислений.

Рассмотрим меры сложности, учитывающие характер разветвлений. В основе узловой меры Вудворда, Хедли лежит идея подсчета топологических характеристик потока управления. При этом, под узловой сложностью понимается число узлов передач управления. Данная мера отслеживает сложность линеаризации программы и чувствительна к структуризации (сложность уменьшается). Она применима для сравнения эквивалентных программ, предпочтительнее меры Холстеда, но по общности уступает мере Мак-Кейба.

Топологическая мера Чена выражает сложность программы числа пересечений границ между областями, образуемыми блок - схемой программы. Этот подход применим только к структурированным программам, допускающим лишь последовательное соединение управляющих конструкций. Для неструктурированных программ мера Чена существенно зависит от условных и безусловных переходов. В этом случае можно указать верхнюю и нижнюю границы меры. Верхняя - есть $m+1$, где m - число логических операторов при их гнездовой вложенности. Нижняя - равна 2. Когда управляющий граф программы имеет только одну компоненту связности, мера Чена совпадает с цикломатической мерой Мак-Кейба.

Метрики Джилба оценивают сложность графоориентированных модулей программ отношением числа переходов по условию к общему числу исполняемых операторов. Хорошо зарекомендовала себя метрика, относящая число межмодульных связей к общему числу модулей. Названные метрики использовались для оценки сложности эквивалентных схем программ, в особенности схем Янова. Используются также меры сложности, учитывающие историю вычислений, характер взаимодействия модулей и комплексные меры.

Совокупность цикломатических мер пригодна для оценивания сложности первичных формализованных спецификаций, задающих в совокупности исходные данные, цели и условия построения искомого ПО. Оценка этой первичной программы или сравнение нескольких альтернативных ее вариантов позволит изначально гармонизировать процесс разработки ПО и от стартовой точки контролировать и управлять его текущей результирующей сложностью.