

# Экономика программной инженерии

Барышникова Марина Юрьевна  
МГТУ им. Н.Э. Баумана  
Каф. ИУ-7

[baryshnikovam@mail.ru](mailto:baryshnikovam@mail.ru)

# Лекция 8

---

## Оценка размера программного продукта на основе метода функциональных точек

Хорошей считается оценка, которая обеспечивает достаточно ясное представление реального состояния проекта и позволяет руководителю проекта принимать хорошие решения относительно того, как управлять проектом для достижения целей

Стив Макконнелл



# Функционально-ориентированные метрики измерения программного продукта

---

- ▶ Функционально-ориентированные метрики косвенно измеряют программный продукт и процесс его разработки
- ▶ При этом рассматривается не размер, а функциональность или полезность продукта
- ▶ В качестве количественной характеристики применяется понятие количества функциональных точек FP (function points)

Для получения функционально-ориентированных метрик (FP-метрик) используются функциональные и концептуальные модели будущей системы (например, модель IDEF0). При этом рассматриваются только наиболее значимые процессы, соответствующие основным функциям разрабатываемого программного продукта (например, перечисленным в техническом задании)



# Метод функциональных точек

---

Функциональная точка — это единица измерения функциональности программного обеспечения. Функциональность программы связана с обработкой информации по запросу пользователя и не зависит от применяемых технических решений.

Пользователи — это отправители и целевые получатели данных, ими могут быть как реальные люди, так и смежные интегрированные информационные системы

Метод функциональных точек позволяет:

- ▶ оценивать категории пользовательских бизнес-функций
- ▶ разрешить проблему, связанную с трудностью получения LOC – оценок на ранних стадиях жизненного цикла
- ▶ определять количество и сложность входных и выходных данных, их структуру, а также внешние интерфейсы, связанные с программной системой



# Метод функциональных точек: историческая справка

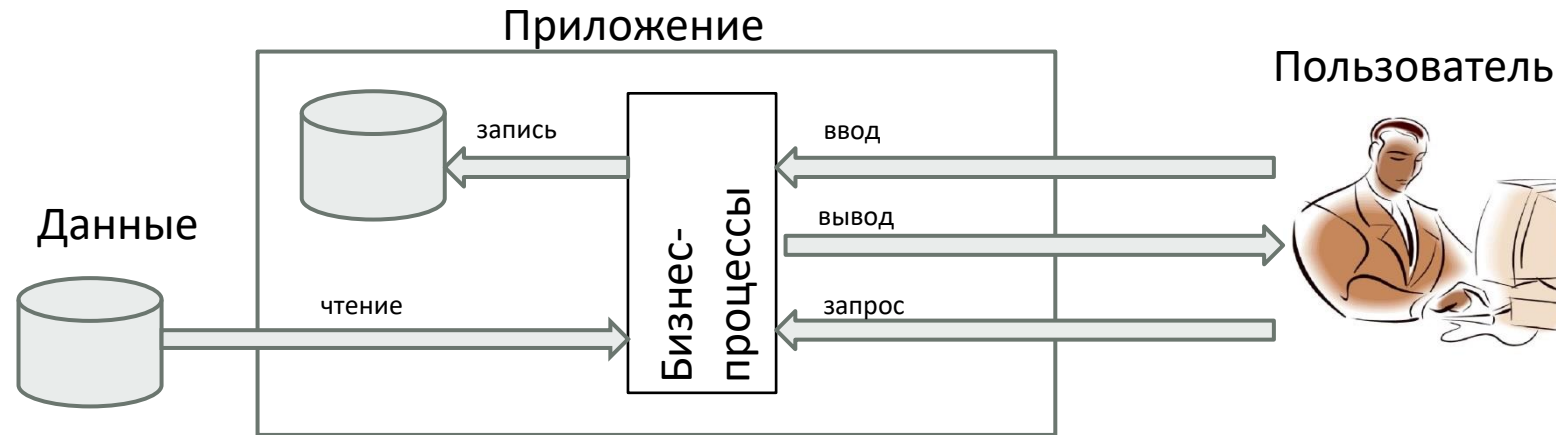
---

- ▶ Функциональные точки (function points) были впервые предложены в качестве метрики измерения размера программного продукта сотрудником компании IBM Аланом Альбрехтом (Allan Albrecht) в 1979 г.
- ▶ Применение функциональных точек основано на изучении требований, поэтому оценка необходимых трудозатрат может быть выполнена на самых ранних стадиях работы над проектом и далее будет уточняться по ходу жизненного цикла
- ▶ Явная связь между требованиями к создаваемой системе и получаемой оценкой позволяет заказчику понять, за что именно он платит, и во что выльется изменение первоначального задания
- ▶ Постепенно метод функциональных точек превратился в индустриальный стандарт, и в 1986 г. для его поддержки и развития была создана некоммерческая организация IFPUG (International Function Point User Group)



# Методика оценки трудоемкости разработки ПО на основе функциональных точек

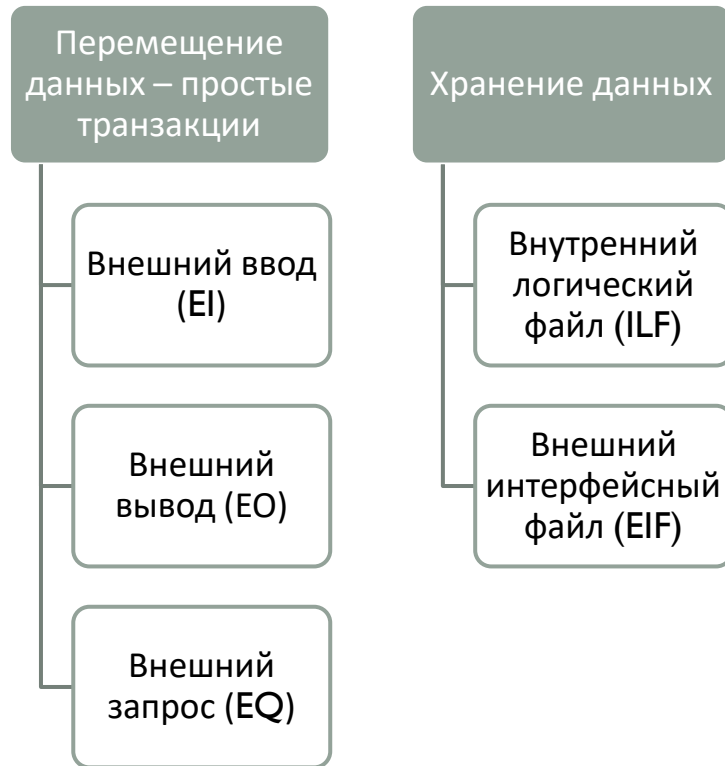
Определение числа функциональных точек является методом количественной оценки ПО, применяемым для измерения функциональных характеристик процессов его разработки и сопровождения независимо от технологии, использованной для его реализации



Трудоемкость вычисляется на основе функциональности разрабатываемой системы, которая, в свою очередь, определяется путем выявления **функциональных типов** — логических групп взаимосвязанных данных, используемых и поддерживаемых приложением, а также **элементарных процессов**, связанных с вводом и выводом информации



# Функциональные типы данных и типы элементарных процессов, используемые в методе функциональных точек



Транзакция — это элементарный процесс, различаемый пользователем и перемещающий данные между внешней средой и программным приложением. В своей работе транзакции используют внутренние и внешние файлы

**Внешний ввод (EI, транзакция, получающая данные от пользователя)** — элементарный процесс, перемещающий данные из внешней среды в приложение. Данные могут поступать с экрана ввода или из другого приложения. Данные могут содержать как управляющую, так и деловую информацию. Обрабатываемые данные могут соответствовать одному или нескольким внутренним логическим файлам

**Внешний вывод (EO, транзакция передающая данные пользователю)** — элементарный процесс, перемещающий данные, вычисленные в приложении, во внешнюю среду и связанный с созданием и/или обработкой выходной информации приложения — выходного отчета, документа, экранной формы

**Внешний запрос (EQ, интерактивный диалог с пользователем, требующий от него каких-либо действий)** — элементарный процесс, состоящий из комбинации «запрос/ответ», не связанный с вычислением производных данных или обновлением внутренних логических файлов (базы данных)

**Внутренний логический файл (ILF, информация, которая используется во внутренних взаимодействиях системы)** — выделяемые пользователем логически связанные группы данных или блоки управляющей информации, которые поддерживаются внутри продукта и обслуживаются через внешние вводы

**Внешний интерфейсный файл (EIF, файлы, участвующие во внешних взаимодействиях с другими системами)** — выделяемые пользователем логически связанные группы данных или блоки управляющей информации, на которые ссылается продукт, но которые поддерживаются вне продукта

# Примеры

---

Информационная характеристика	Элементы данных
Внешние вводы	Поля ввода данных, сообщения об ошибках, вычисляемые значения, кнопки
Внешние выводы	Поля данных в отчетах, вычисляемые значения, сообщения об ошибках, заголовки столбцов, которые читаются из внутреннего файла
Внешние запросы	Вводимые элементы: поле, используемое для поиска, щелчок мыши. Выводимые элементы — отображаемые на экране поля





# Информационные характеристики, используемые в методе функциональных точек

---

- ▶ *Количество внешних вводов.* Подсчитываются все вводы пользователя, по которым поступают разные прикладные данные, при этом вводы должны быть отделены от запросов, которые подсчитываются отдельно
- ▶ *Количество внешних выводов.* Подсчитываются все выводы, по которым к пользователю поступают результаты, вычисленные программным приложением. В этом контексте выводы означают отчеты, экраны, распечатки, сообщения об ошибках. Индивидуальные единицы данных внутри отчета отдельно не подсчитываются
- ▶ *Количество внешних запросов.* Под запросом понимается диалоговый ввод, который приводит к немедленному программному ответу в форме диалогового вывода. При этом диалоговый ввод в приложении не сохраняется, а диалоговый вывод не требует выполнения вычислений. Подсчитываются все запросы — каждый учитывается отдельно
- ▶ *Количество внутренних логических файлов.* Подсчитываются все логические файлы (то есть логические группы данных, которые могут быть частью базы данных или отдельным файлом)
- ▶ *Количество внешних интерфейсных файлов.* Подсчитываются все логические файлы из других приложений, на которые ссылается данное приложение



# Как связаны информационные характеристики?

---

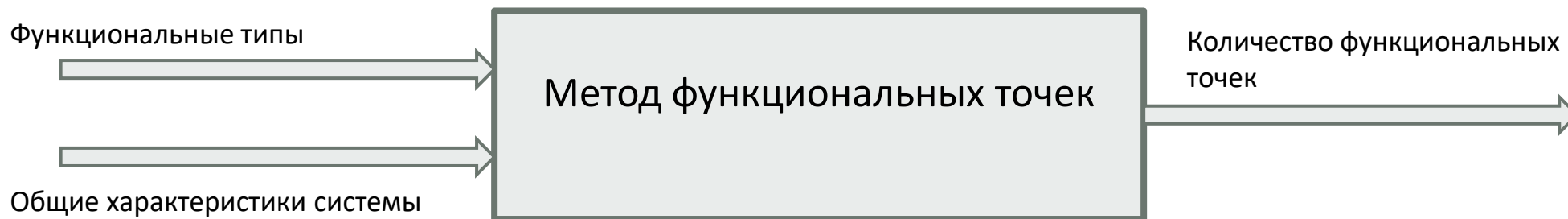
- ▶ ILF и EIF являются логически связанными группами данных (файлами, таблицами баз данных и т.п.), определяемыми пользователем и необходимыми для работы рассматриваемого программного средства. Оба эти типа файлов могут использоваться при генерации внешних выводов (EO), а также при обслуживании внешних запросов (EQ). Разница между ними состоит лишь в том, что ILF поддерживаются самим рассматриваемым ПС путем использования внешних вводов (EI), а EIF поддерживаются каким-либо другим приложением
- ▶ Функциональность логических файлов (ILF и EIF) оценивается путем подсчета количества типов элементов записей (RET) и количества типов элементов данных (DET), входящих в соответствующие логические группы данных. При этом под количеством RET обычно понимается количество различных логических подгрупп данных, выделяемых в файле с точки зрения пользователя, или количество различных используемых форматов записей, а под количеством DET - количество различных элементарных полей в этих записях
- ▶ Основным источником информации для оценки количества RET и DET в логическом файле могут служить, например, словарь данных (Data Dictionary), составленный в ходе моделирования информационной системы с использованием диаграмм потоков данных, логическая структура баз данных, полученная при построении моделей «сущность-связь», логическая структура файлов или баз данных, поддерживаемых другими приложениями и описанная в их документации (для EIF), и т.п.



# Порядок расчета трудоемкости разработки ПО

---

- ▶ определение количества функциональных типов приложения
- ▶ определение количества связанных с каждым функциональным типом элементарных данных (DET), элементарных записей (RET) и файлов типа ссылок (FTR)
- ▶ определение сложности (в зависимости от количества DET, RET и FTR)
- ▶ подсчет количества функциональных точек приложения
- ▶ подсчет количества функциональных точек с учетом общих характеристик системы
- ▶ оценка трудоемкости разработки (с использованием различных статистических данных)



# Определение количества и сложности транзакционных функциональных типов

---

- ▶ Количество транзакционных функциональных типов (входных элементов приложения, выходных элементов приложения и внешних запросов) определяется на основе выявления входных и выходных документов, экранных форм, отчетов, а также по диаграммам классов
- ▶ Далее для каждого выявленного функционального типа (EI, EO или EQ) определяется его сложность (низкая, средняя или высокая), которая зависит от количества связанных с этим функциональным типом DET, RET и FTR

## Еще раз про DET, RET, FTR

- ▶ DET – это уникальное распознаваемое пользователем, нерекурсивное (неповторяющееся) поле данных
- ▶ RET - идентифицируемая пользователем логическая группа данных внутри ILF или EIF
- ▶ FTR – это тип файла, на который ссылается транзакция. FTR позволяет подсчитать количество различных файлов (информационных объектов) типа ILF и/или EIF модифицируемых или считываемых в транзакции



# Ранг и оценка сложности внутренних логических файлов

Типы элементов-записей (RET)	Элементы данных (DET)		
	1-19	20-50	>50
1	Низкий (7)	Низкий (7)	Средний (10)
2-5	Низкий (7)	Средний (10)	Высокий (15)
>5	Средний (10)	Высокий (15)	Высокий (15)

## Ранг и оценка сложности внешних интерфейсных файлов

Типы элементов-записей (RET)	Элементы данных (DET)		
	1-19	20-50	>50
1	Низкий (5)	Низкий (5)	Средний (7)
2-5	Низкий (5)	Средний (7)	Высокий (10)
>5	Средний (7)	Высокий (10)	Высокий (10)



# Правила расчета DET для внешних вводов

В качестве DET (data element types) для внешних вводов (EI) учитываются:

- ▶ каждое нерекурсивное поле, принадлежащее внутреннему логическому файлу (ILF) (или поддерживаемое им) и обрабатываемое во вводе
- ▶ каждое поле, которое пользователь хотя и не вызывает, но оно через процесс ввода поддерживается во внутреннем логическом файле (ILF)
- ▶ логическое поле, которое физически представляет собой множество полей, но воспринимается пользователем как единый блок информации
- ▶ группа полей, которые появляются во внутреннем логическом файле (ILF) более одного раза, но в связи с особенностями алгоритма их использования воспринимаются как один DET
- ▶ группа полей, которые фиксируют ошибки в процессе обработки или подтверждают, что обработка закончилась успешно
- ▶ действие, которое может быть выполнено во вводе

## Ранг и оценка сложности внешних вводов

Ссылки на файлы (FTR)	Элементы данных (DET)		
	1-4	5-15	>15
0-1	Низкий (3)	Низкий (3)	Средний (4)
2	Низкий (3)	Средний (4)	Высокий (6)
>2	Средний (4)	Высокий (6)	Высокий (6)



# Правила расчета DET для внешних выводов

В качестве DET для внешних выводов (ЕО) учитываются:

- ▶ каждое распознаваемое пользователем нерекурсивное поле, участвующее в процессе вывода
- ▶ поле, которое физически отображается в виде нескольких полей его составляющих, но используется как единый информационный элемент
- ▶ каждый тип метки и каждое значение числового эквивалента при графическом выводе
- ▶ текстовая информация, которая может содержать одно слово, предложение или фразу

Примечание: переменные, определяющие номера страниц или генерируемые системой логотипы не являются элементами данных

## Ранг и оценка сложности внешних выводов

Ссылки на файлы (FTR)	Элементы данных (DET)		
	1-4	5-19	>19
0-1	Низкий (4)	Низкий (4)	Средний (5)
2-3	Низкий (4)	Средний (5)	Высокий (7)
>3	Средний (5)	Высокий (7)	Высокий (7)



# Правила расчета DET для внешних запросов

---

В качестве DET для внешнего запроса (EQ) **по входу** учитываются:

- ▶ каждое распознаваемое пользователем нерекурсивное поле, появляющееся во вводной части запроса
- ▶ каждое поле, которое определяет критерий выбора данных
- ▶ группа полей, в которых выдаются сообщения о возникающих ошибках в процессе ввода информации или подтверждающих успешное завершение процесса ввода
- ▶ группа полей, которые позволяют выполнять запросы

В качестве DET для внешнего запроса (EQ) **по выходу** учитываются:

- ▶ каждое распознаваемое пользователем нерекурсивное поле, которое появляется в выводной части запроса
- ▶ логическое поле, которое физически отображается как группа полей, однако воспринимается пользователем как единое поле
- ▶ группа полей, которые в соответствии с методикой обработки могут повторяться во внутреннем логическом файле (ILF)

Примечание: колонтитулы или генерируемые системой иконки не учитываются как DET

---





# Ранг и оценка сложности внешних запросов

---

Ссылки на файлы (FTR)	Элементы данных		
	1-4	5-19	>19
0-1	Низкий (3)	Низкий (3)	Средний (4)
2-3	Низкий (3)	Средний (4)	Высокий (6)
>3	Средний (4)	Высокий (6)	Высокий (6)



# Учет элементов данных из графического интерфейса пользователя

---

Элемент данных	Правило учета
Группа радиокнопок	Так как в группе пользователь выбирает только одну радиокнопку, все радиокнопки группы считаются одним элементом данных
Группа флажков (переключателей)	Так как в группе пользователь может выбрать несколько флажков, каждый флажок считают элементом данных
Командные кнопки	Командная кнопка может определять действие добавления, изменения или запроса. Кнопка ОК может вызывать транзакции различных типов. Каждая кнопка считается отдельным элементом данных
Списки	Список может быть внешним запросом, но результат запроса может быть элементом данных внешнего ввода



# Алгоритм применения метода функциональных точек

---

- ▶ Подсчитываются функции для каждой категории (вводы, выводы, запросы, структуры данных, интерфейсы)
- ▶ Устанавливаются требования для каждой категории
- ▶ Определяется сложность каждой функции (высокая, средняя, низкая)
- ▶ Каждая функция умножается на соответствующий ей параметр, а затем суммируется с целью получения общего количества функциональных точек

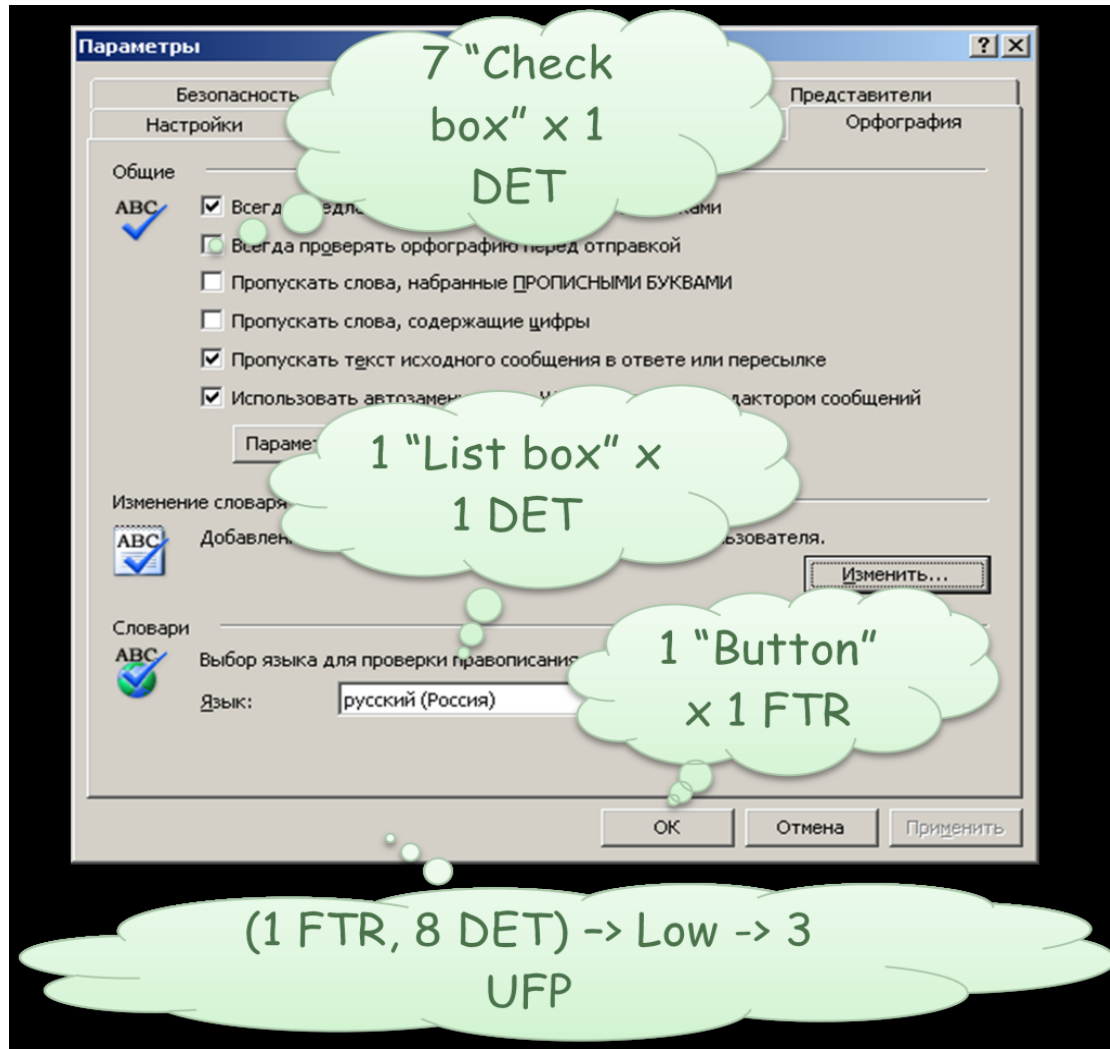


# Исходные данные для расчета общего количества функциональных точек

Имя характеристики	Ранг, сложность, количество			
	Низкий	Средний	Высокий	Итого
Внешние вводы	__x3 = __	__x4 = __	__x6 = __	= __
Внешние выводы	__x4 = __	__x5 = __	__x7 = __	= __
Внешние запросы	__x3 = __	__x4 = __	__x6 = __	= __
Внутренние логические файлы	__x7 = __	__x 1__ = __	__x15 = __	= __
Внешние интерфейсные файлы	__x5 = __	__x7 = __	__x1__ = __	= __
Общее количество				= __



# Диалоговое окно, управляющее проверкой орфографии в MS Office Outlook



Источник: С. Архипенков. Лекции по управлению программными проектами, стр. 102

# Примеры анализа данных

---

Проектируется GUI для обслуживания клиентов, который будет иметь поля: *Имя, Адрес, Город, Страна, Почтовый Индекс, Телефон, Email* (таким образом, имеется 7 полей или 7 DET)

Кроме того, предусматриваются 3 командные кнопки: *Добавить, Изменить, Удалить*. Следовательно каждый из внешних вводов *Добавить, Изменить, Удалить* будет состоять из 8 DET (7 полей плюс командная кнопка)

В данном приложении генерируются 3 типа сообщений: *сообщения об ошибке, сообщения подтверждения и сообщения уведомления*. Сообщения об ошибке (например, *Требуется пароль*) и сообщения подтверждения (например, *Вы действительно хотите удалить клиента?*) указывают, что произошла ошибка или что процесс может быть завершен. Эти сообщения не образуют самостоятельного процесса, они являются частью другого процесса, то есть считаются элементом данных соответствующей транзакции

Уведомление является независимым элементарным процессом. Например, при попытке получить из банкомата сумму денег, превышающую их количество на счете, генерируется сообщение *Не хватает средств для завершения транзакции*. Оно является результатом чтения информации из файла счета и формирования заключения. Сообщение уведомления рассматривается как внешний вывод



# Скорректированное количество функциональных точек

---

$FP = \text{Общее количество} * (0,65 + 0,01 * \sum Fi),$

где  $Fi$  — 14 коэффициентов регулировки сложности

Каждый коэффициент может принимать следующие значения:

- 0 — нет влияния,
- 1 — случайное,
- 2 — небольшое,
- 3 — среднее,
- 4 — важное,
- 5 — основное



# Определение системных параметров приложения

№	Системный параметр	Описание
1	Передача данных	Сколько средств связи требуется для передачи или обмена информацией с приложением или системой?
2	Распределенная обработка данных	Как обрабатываются распределенные данные и функции обработки?
3	Производительность	Нуждается ли пользователь в фиксации времени ответа или производительности?
4	Эксплуатационные ограничения	Насколько сильны эксплуатационные ограничения и каков объем специальных усилий на их преодоление?
5	Частота транзакций	Как часто выполняются транзакции (каждый день, каждую неделю, каждый месяц) ?
6	Оперативный ввод данных	Какой процент информации надо вводить в режиме онлайн?
7	Эффективность работы конечных пользователей	Приложение проектировалось для обеспечения эффективной работы конечного пользователя?
8	Оперативное обновление	Как много внутренних файлов обновляется в онлайн-транзакции?
9	Сложность обработки	Выполняет ли приложение интенсивную логическую или математическую обработку?
10	Повторная используемость	Приложение разрабатывалось для удовлетворения требований одного или многих пользователей?
11	Легкость инсталляции	Насколько трудны преобразование и инсталляция приложения?
12	Легкость эксплуатации	Насколько эффективны и/или автоматизированы процедуры запуска, резервирования и восстановления?
13	Количество возможных установок на различных платформах	Была ли спроектирована, разработана и поддержана возможность инсталляции приложения в разных местах для различных организаций?
14	Простота изменений (гибкость)	Была ли спроектирована, разработана и поддержана в приложении простота изменений?



# Передачи данных

---

Значение	Описание
0	Полностью пакетная обработка на локальном ПК
1	Пакетная обработка, удаленный ввод данных или удаленная печать
2	Пакетная обработка, удаленный ввод данных и удаленная печать
3	Сбор данных в режиме «онлайн» или дистанционная обработка, связанная с пакетным процессом
4	Несколько внешних интерфейсов, один тип коммуникационного протокола
5	Несколько внешних интерфейсов, несколько типов коммуникационных протоколов



# Распределенная обработка данных

---

Значение	Описание
0	Передача данных или процессов между компонентами системы отсутствует
1	Приложение готовит данные для обработки на ПК конечного пользователя
2	Данные готовятся для передачи, затем передаются и обрабатываются на другом компоненте системы (не на ПК конечного пользователя)
3	Распределенная обработка и передача данных в режиме «онлайн» только в одном направлении
4	Распределенная обработка и передача данных в режиме «онлайн» в обоих направлениях
5	Динамическое выполнение процессов в любом подходящем компоненте системы



# Производительность

---

Значение	Описание
0	К системе не предъявляется специальных требований, касающихся производительности
1	Требования к производительности определены, но не требуется никаких специальных действий
2	Время реакции или пропускная способность являются критическими в пиковые периоды. Не требуется никаких специальных решений относительно использования ресурсов процессора. Обработка может быть завершена в течение следующего рабочего дня
3	Время реакции или пропускная способность являются критическими в обычное рабочее время. Не требуется никаких специальных решений относительно использования ресурсов процессора. Время обработки ограничено взаимодействующими системами
4	То же, кроме того, пользовательские требования к производительности достаточно серьезны, чтобы ее необходимо было анализировать на стадии проектирования
5	То же, кроме того, на стадиях проектирования, разработки и (или) реализации для удовлетворения пользовательских требований к производительности используются специальные средства анализа



# Эксплуатационные ограничения

---

Значение	Описание
0	Какие-либо явные или неявные ограничения отсутствуют
1	Эксплуатационные ограничения присутствуют, но не требуют никаких специальных усилий
2	Должны учитываться некоторые ограничения, связанные с безопасностью или временем реакции
3	Должны учитываться конкретные требования к процессору со стороны конкретных компонентов приложения
4	Заданные эксплуатационные ограничения требуют специальных ограничений на выполнение приложения в центральном или выделенном процессоре
5	То же, кроме того, специальные ограничения затрагивают распределенные компоненты системы



# Частота транзакций

---

Значение	Описание
0	Пиковых периодов не ожидается
1	Ожидаются пиковые периоды (ежемесячные, ежеквартальные, ежегодные)
2	Ожидаются еженедельные пиковые периоды
3	Ожидаются ежедневные пиковые периоды
4	Высокая частота транзакций требует анализа производительности на стадии проектирования
5	То же, кроме того, на стадиях проектирования, разработки и (или) внедрения необходимо использовать специальные средства анализа производительности



# Оперативный ввод данных

---

Значение	Описание
0	Все транзакции обрабатываются в пакетном режиме
1	От 1% до 7% транзакций требуют интерактивного ввода данных
2	От 8% до 15% транзакций требуют интерактивного ввода данных
3	От 16% до 23% транзакций требуют интерактивного ввода данных
4	От 24% до 30% транзакций требуют интерактивного ввода данных
5	Более 30% транзакций требуют интерактивного ввода данных



# Эффективность работы конечных пользователей

---

Эффективность работы конечных пользователей определяется наличием следующих функциональных возможностей:

- ▶ Средства навигации (например, функциональные клавиши, динамически генерируемые меню)
- ▶ Меню
- ▶ Онлайн-подсказки и документация
- ▶ Автоматическое перемещение курсора
- ▶ Скроллинг
- ▶ Удаленная печать
- ▶ Предварительно назначенные функциональные клавиши
- ▶ Выбор данных на экране с помощью курсора
- ▶ Использование видеоэффектов, цветового выделения, подчеркивания и других индикаторов
- ▶ Всплывающие окна
- ▶ Минимизация количества экранов, необходимых для выполнения бизнес-функций
- ▶ Поддержка двух и более языков



# Эффективность работы конечных пользователей

---

Значение	Описание
0	Ни одной из перечисленных функциональных возможностей
1	От одной до трех функциональных возможностей
2	От четырех до пяти функциональных возможностей
3	Шесть или более функциональных возможностей при отсутствии конкретных пользовательских требований к эффективности
4	То же, кроме того, пользовательские требования к эффективности требуют специальных проектных решений для учета эргономических факторов (например, минимизации нажатий клавиш, максимизации значений по умолчанию, использования шаблонов)
5	То же, кроме того, пользовательские требования к эффективности требуют применения специальных средств и процессов, демонстрирующих их выполнение





# Оперативное обновление

---

Значение	Описание
0	Отсутствует
1	Онлайновое обновление от одного до трех управляющих файлов Объем обновлений незначителен, восстановление несложно
2	Онлайновое обновление четырех или более управляющих файлов Объем обновлений незначителен, восстановление несложно
3	Онлайновое обновление основных внутренних логических файлов
4	То же, плюс необходимость специальной защиты от потери данных
5	То же, кроме того, большой объем данных требует учета затрат на процесс восстановления. Требуются автоматизированные процедуры восстановления с минимальным вмешательством оператора



# Сложность обработки

---

Сложность обработки характеризуется наличием у приложения следующих функциональных возможностей:

- ▶ повышенная реакция на внешние воздействия и (или) специальная защита от внешних воздействий
- ▶ экстенсивная логическая обработка
- ▶ экстенсивная математическая обработка
- ▶ обработка большого количества исключительных ситуаций
- ▶ поддержка разнородных типов входных/выходных данных

Значение	Описание
0	Ни одной из перечисленных функциональных возможностей
1	Любая одна из возможностей
2	Любые две возможности
3	Любые три возможности
4	Любые три возможности
5	Все пять возможностей



# Повторное использование

---

Значение	Описание
0	Отсутствует
1	Повторное использование кода внутри одного приложения
2	Не более 10% приложений будут использоваться более чем одним пользователем
3	Более 10% приложений будут использоваться более чем одним пользователем
4	Приложение оформляется как продукт и (или) документируется для облегчения повторного использования. Настройка приложения выполняется пользователем на уровне исходного кода
5	То же, с возможностью параметрической настройки приложений



# Легкость инсталляции

---

Значение	Описание
0	К установке не предъявляется никаких специальных требований
1	Для установки требуется специальная процедура
2	Заданы пользовательские требования к конвертированию (переносу существующих данных и приложений в новую систему) и установке, должны быть обеспечены и проверены соответствующие руководства. Конвертированию не придается важное значение
3	То же, однако конвертированию придается важное значение
4	То же, что и в случае 2, плюс наличие автоматизированных средств конвертирования и установки
5	То же, что и в случае 3, плюс наличие автоматизированных средств конвертирования и установки



# Легкость эксплуатации

---

Значение	Описание
0	К эксплуатации не предъявляется никаких специальных требований, за исключением обычных процедур резервного копирования
1-4	Приложение обладает одной, несколькими или всеми из перечисленных далее возможностей. Каждая возможность, за исключением второй, обладает единичным весом: 1) наличие процедур запуска, копирования и восстановления с участием оператора; 2) то же, без участия оператора; 3) минимизируется необходимость в монтировании носителей для резервного копирования; 4) минимизируется необходимость в средствах подачи и укладки бумаги при печати
5	Вмешательство оператора требуется только при запуске и завершении работы системы. Обеспечивается автоматическое восстановление работоспособности приложения после сбоев и ошибок



# Количество возможных установок на различных платформах

---

Значение	Описание
0	Приложение рассчитано на установку у одного пользователя
1	Приложение рассчитано на много установок для строго стандартной платформы (технические средства + программное обеспечение)
2	Приложение рассчитано на много установок для платформ с близкими характеристиками
3	Приложение рассчитано на много установок для различных платформ
4	То же, что в случаях 1 или 2, плюс наличие документации и планов поддержки всех установленных копий приложения
5	То же, что в случае 3, плюс наличие документации и планов поддержки всех установленных копий приложения



# Простота изменений (гибкость)

Гибкость характеризуется наличием у приложения следующих возможностей:

- ▶ поддержка простых запросов, например, логики и (или) в применении только к одному внутреннему логическому файлу (ILF) (вес — 1)
- ▶ поддержка запросов средней сложности, например, логики и (или) в применении более чем к одному ILF (вес - 2)
- ▶ поддержка сложных запросов, например, комбинации логических связей и (или) в применении к одному или более ILF (вес — 3)
- ▶ управляющая информация хранится в таблицах, поддерживаемых пользователем в интерактивном режиме, однако эффект от ее изменений проявляется на следующий рабочий день
- ▶ то же, но эффект проявляется немедленно (вес — 2)

Значение	Описание
0	Ни одной из перечисленных функциональных возможностей
1	Любая одна из возможностей
2	Любые две возможности
3	Любые три возможности
4	Любые три возможности
5	Все пять возможностей



# Пересчет FP-оценок в LOC-оценки

Язык программирования	Количество операторов на одну FP
Ассемблер	320
C	128
Кобол	106
Фортран	106
Паскаль	90
C++	64
C#	53
Java	53

Язык программирования	Количество операторов на одну FP
Ada 95	49
Visual Basic	32
Visual C++	34
Delphi 5	18
Perl	21
Prolog	54
ANSI SQL	13
Java Script	56





# Размер программного обеспечения в FP и LOC

Тип ПО	Размер, FP	Размер, LOC	Количество LOC на одну FP
Текстовые процессоры	3500	437500	125
Электронные таблицы	3500	437500	125
Клиент-серверные приложения	7500	675000	90
ПО баз данных	7500	937500	125
Производственные приложения	7500	937500	125
Крупные бизнес-приложения	10000	1050000	105
Операционные системы	75000	11250000	150
Системы масштаба предприятий	150000	18750000	125
Крупные оборонные системы	250000	25000000	100



# Распределение временных затрат по стадиям для маленьких и больших проектов

Масштаб проекта	Начальная стадия, %	Проектирование, %	Разработка, %	Ввод в действие, %
Маленький коммерческий проект	10	20	50	20
Большой, сложный проект	15	30	40	15



# Статистические данные по времени разработки

Размер проекта	< 100 FP	100 - 1000 FP	1000 - 10000 FP	> 10000 FP
Планируемый срок (мес.)	6	12	18	24
Реальный срок (мес.)	8	16	24	36
Отставание	2	4	6	12

Примечание: причины отставания частично объясняются неточной оценкой, частично — ростом количества требований к системе после того, как выполнена начальная оценка



# Производительность разработчиков для стандартных типов проектов

Тип программы	Строк кода на человеко-месяц (номинал)		
	проект на 1000 LOC	проект на 10000 LOC	проект на 25000 LOC
Авиационное оборудование	100 - 1000 (200)	20 - 300 (50)	20 - 200 (40)
Бизнес-система	800 - 18000 (3000)	200 - 7000 (600)	100 - 5000 (500)
Системы управления	200 - 3000 (500)	50 - 600 (100)	40 - 500 (80)
Встроенные системы	100 - 2000 (300)	30 - 500 (70)	20 - 400 (60)
Открытые Интернет-системы	600 - 10000 (1500)	100 - 2000 (300)	100 - 1500 (200)
Микрокод	100 - 800 (200)	20 - 200 (40)	20 - 100 (30)
Управление процессами	500 - 5000 (1000)	100 - 1000 (300)	80 - 900 (200)
Системы реального времени	100 - 1500 (200)	20 - 300 (50)	20 - 300 (40)
Системы научных и инженерных исследований	500 - 7500 (1000)	100 - 1500 (300)	80 - 1000 (200)
Коммерческие пакеты	400 - 5000 (1000)	100 - 1000 (200)	70 - 800 (200)
Системные программы/драйверы	200 - 5000 (600)	50 - 1000 (100)	40 - 800 (90)
Телекоммуникации	200 - 3000 (600)	50 - 600 (100)	40 - 500 (90)



# Основные выводы

---

- ▶ Оценка трудоемкости должна быть вероятностным утверждением. Это означает, что для нее существует некоторое распределение вероятности, которое может быть очень широким, если неопределенность высокая, или достаточно узким, если неопределенность низкая
- ▶ Использование собственного опыта или опыта коллег, полученного в похожих проектах, это наиболее прагматичный подход, который позволяет получить достаточно реалистичные оценки трудоемкости и срока реализации программного проекта, быстро и без больших затрат
- ▶ Если собственный опыт аналогичных проектов отсутствует, а экспертное мнение недоступно, то необходимо использовать формальные методики, основанные на обобщенном отраслевом опыте, а именно метод функциональных точек или модель COSOMO II
- ▶ Недооценка приводит к ошибкам планирования и неэффективному взаимодействию. Агрессивные сроки, постоянное давление, сверхурочные, авралы служат причиной того, что затраты на проект растут экспоненциально и неограниченно





Спасибо за внимание!