

# Оценки моделей процессов в IT

## §1. Измерение меры и метрики в IT

**Качество ПО** — совокупность свойств, определяющих полезность этого ПО для пользователей в соответствии с функциональным назначением и предъявленными требованиями.

*(скорее всего не будет в рубрике, если будет, то только на описательном уровне)*

**Характеристика качества программы** — понятие, отражающее отдельные факторы, влияющие на качество программы и поддающиеся измерению.

**Критерий качества ПО** — это численный показатель, характеризующий степень, в которой программе присуще оцениваемое свойство.

**Требования к критерию:**

- 1) Численно характеризовать основную целевую функцию программы
- 2) Обеспечить возможность определения затрат, необходимых для достижения требуемого уровня качества, а также степени влияния на показатель качества различных внешних факторов.
- 3) Быть по возможности простым, хорошо измеримым, иметь малую дисперсию

Для измерения характеристики критериев качества ПО используют метрики.

*(содержательно)* **Метрики качества ПО** — системы измерения качества ПО.

*(определение по IEEE)* **Метрика** — мера степени обладания свойством, имеющая численное значение.

Метрика  $\approx$  мера  $\approx$  критерий.

В исследовании метрик различают два направления:

- 1) Поиск (изобретение метрик), характеризующих свойства ПО — поиск метрик оценки самого ПО
- 2) Поиск метрик для оценки условий разработки программ (технических характеристик и факторов разработки ПО)

По виду информации, получаемой при оценке качества ПО:

- 3) Метрики, оценивающие отклонение от нормы характеристик исходных проектных материалов
- 4) Метрики прогноза качества разрабатываемого ПО
- 5) Метрики соответствия разработки заданным требованиям

Качественные оценки ПО:

- 1) Оценки топологической и информационной сложности программ
- 2) Оценки надежности программных систем для прогнозирования отказов
- 3) Оценки производительности ПО, повышение его эффективности путем выявления ошибок проектирования
- 4) Оценки качества и уровня применения языковых средств
- 5) Оценки трудности восприятия и понимания программных текстов
- 6) Оценка производительности труда программистов для прогнозирования сроков разработки программ

## §2. Метрики для оценки архитектуры ПО

Мы рассматриваем только метрики, связанные с качеством построения информационных моделей, но это метрики в свою очередь могут задаваться на 5 уровнях:

- 1) Предметная область
- 2) Модель предметной области
- 3) Логическая модель данных
- 4) Физическая модель данных
- 5) Собственно БД и приложения

Для архитектурных моделей ПО используются классические проектные характеристики:

- 1) **Модульность**: модуль — фрагмент программного кода, являющийся «строительным блоком» для физической структуры системы. Как правило, модуль состоит из интерфейсной части и части реализации. Модульность — свойство системы, которое позволяет декомпозировать ее на набор внутренне связанных и слабо зависящих друг от друга модулей. Она позволяет создать сколь угодно сложную программу. Идея модульности основана на тезисе «Разделяй и властвуй!»: сложную проблему легче решить, разделяя ее на управляемые части, то есть модульность — свойство интеллектуальной разработки программы. Однако при бездумном применении принципа модульности возрастают затраты на межмодульный интерфейс. В принципе для каждой программной разработки существует оптимальное кол-во модулей, минимизирующее стоимость разработки, однако теоретического обоснования для такой оптимизации нет. Существует оценочная эвристика для оценки перспективности модуля (два критерия):

- 1) Оптимальный модуль снаружи проще, чем внутри
- 2) Оптимальные модуль проще использовать, чем построить

- 2) **Информационная закрытость** — содержание модулей должно быть скрыто друг от друга, то есть модуль должен проектироваться так, чтобы его содержимое (процедуры и данные) было не доступно модулям клиентов, которые не нуждаются в этой информации. Достоинства:

- 1) Обеспечивается возможность разработки модулей различными независимыми коллективами
- 2) Обеспечивается легкая модификация системы за счет малого распространения ошибок

Предполагается два типа связей: внутренняя и внешняя, а идеальная программная архитектура — это набор черных ящиков, которые связаны друг с другом минимальным набором однотипных связей

- 3) **Связность** — внутренняя характеристика модуля (мера зависимости частей). Существует 7 типов связности:

- 1) Связность по совпадению ( $CC=0$ ) — отсутствуют явно выраженные внутренние связи. Причины возникновения такой связности:
  - 1) Бездумный перевод существующего монолитного кода в модули
  - 2) Необоснованные изменения модулей с плохой временной связностью, приводящие к добавлению флажков.
- 2) Логическая связность ( $CC=1$ ) — части модуля объединяются по принципу функционального подобия. Такой модуль имеет «уродливый внешний вид» с различного типа параметрами, обеспечивающие разные типы доступа и запутанную внутреннюю структуру типа лабиринта.
- 3) Временная связность ( $CC=3$ ) — части модуля не связаны, но необходимы в один и тот же период работы системы. Сильная взаимосвязь с другими модулями => сильная чувствительность к внеению изменений.
- 4) Процедурная связность ( $CC=5$ ) — части модуля связаны порядком выполняемых ими действий, реализующих некоторый сценарий поведения.

Важность ошибки определяется тем, сколько стоит поддерживаемая процедура.

- 5) Коммуникативная связность ( $CC=7$ ) — части модуля связаны по данным (работают с одной и той же структурой данных). Проблема: избыточность получаемых результатов (то есть ухудшается сопровождение модуля — дублирование и т. п.)
- 6) Информационная связность ( $CC=9$ ) — выходные данные одной части используются как входные данные в другой части модуля.
- 7) Функциональная связность ( $CC=10$ ) — модуль содержит элементы, участвующие в выполнении одной и только одной, проблемной функции (единой с точки зрения клиента).

Аналогичные проблемы возникают при структурировании бизнес-функции в организации.

- 4) **Сцепление** — мера взаимозависимости модулей по данным. Сцепление нужно уменьшать.

- 1) Сцепление по данным ( $CC=1$ ) — модуль А вызывает модуль В.
- 2) Сцепление по образцу ( $CC=3$ ) — в качестве параметров используются структуры данных
- 3) Сцепление по управлению ( $CC=4$ ) — управляющие данные посылаются в какой-то модуль, управление происходит с помощью флагов или переключателей.
- 4) Сцепление по внешним ссылкам ( $CC=5$ ) — оба модуля ссылаются на один и тот же глобальный элемент данных.
- 5) Сцепление по общей области ( $CC=7$ ) — модули разделяют одну и ту же глобальную структуру данных.
- 6) Сцепление по содержанию ( $CC=9$ ) — один модуль прямо ссылается на содержание другого не через точки входа (напр., коды команд смежаются друг с другом).

- 5) **Сложность**. В простейшем случае сложность системы определяется как сумм мер сложностей модулей. Сложность модуля вычисляется по-разному:

- 1) Мера длины модуля (число различных операторов и числа различных операндов)
- 2) Объем модуля (пропорционально кол-ву символов для записи всех операторов и операндов)
- 3) Учет топологии внутренних связей — метрика цикломатической сложности:  
 $V = E - N + 2$  ( $E$  — кол-во дуг,  $N$  — кол-во вершин в управляющем графе)

Дальнейшее уточнение оценок сложности предполагает, что каждый модуль рассматривается как локальная структура графового типа.

Таким образом, при комплексной оценке сложности программы рассматриваем следующие хар-ки:

- 1) Мера сложности модулей
- 2) Лог-оценка (кол-во строк в программном продукте)
- 3) Сложность приложения (в первую очередь кол-во транзакций)
- 4) Добавляем меру сцепления
- 5) Добавляем меру связности
- 6) Добавляем хар-ки управления (хар-ки управления программной системы)

**Локальными характеристиками модуля структуры** — это коэффициент объединения по входа и коэффициент разветвления по выходу.

**Коэффициент объединения по входу** — количество модулей, которые прямо управляют данным модулем (он должен быть как можно меньше, чтобы система была устойчивее).

**Коэффициент разветвления по выходу** — количество модулей, которыми прямо управляет этот модуль.

Из практики проектирования известно, что лучшее решение обеспечивается иерархической структурой в виде дерева. Степень отличия реальной проектной структуры от дерева характеризуется невязкой структуры.

**Невязка** — позиция данного графа между деревом и полным графом.

Если у полного графа  $n$  вершин, то кол-во ребер  $e_s = n * (n - 1) / 2$

Если у дерева  $n$  вершин, то кол-во ребер  $e_t = n - 1$

Формулу невязки можно построить, сравнивая кол-во ребер полного графа, реального графа и дерева.

Невязка:  $N_{ev} = (e - e(r)) / (e - e(t))$  — значение лежит от 0 до 1.

Невязка — очень грубая оценка структуры.

Для увеличения точности оценки нужно добавить характеристики связности и сцепления, а именно: хорошая структура должна иметь меньшее сцепление и большую связность.

Такие коэффициенты предложили Константаин и Йордан (надо заполнить, они предложили паттерны проектирования). Они ввели два коэффициента, которые связаны с управляющими потоками, то есть кол-вом вызовов модулей. Коэффициенты Константаина и Йордана не учитывают веса ребер (то есть частоту вызовов).

**Коэффициент Генри** — информационный коэффициент, который учитывает количество элементов и структур данных, из которых модуль берет информацию и которые он обновляет.

Берутся эти два коэффициенты, и на их основании вычисляется метрика общей сложности структуры, приведенная к числу строк кода модуля.

На сегодня меры сложности графа находят широкое применение: например, оценка качества антологии (графа, описывающие предметную область), оценка качества концепт-графов, оценки качества графов-планов (например, учебных программ).

## §3. Метрики сопровождения процесса разработки ПО

Первое, что мы спрашиваем: зачем нам нужны метрики для сопровождения проекта. Возможны два варианта:

- 1) Анализ и прогнозирование рисков по проекту. Используется 3 способа:
  - 1) Анализ прошлых аналогичных проектов (отвечаем на вопрос: чем мы отличаемся от них, как в худшую, так и в лучшую сторону)
  - 2) Мозговой штурм (вся проектная команда приглашается на общее совещание и каждый озвучивает риски, которые он видит в проекте)
  - 3) Использование стандартного списка риска (нужно озаботиться тем, чтобы эти списки найти: например, есть стандарт (ГОСТ), в котором есть методики оценки риска)
- 2) Для анализа состояния вводятся метрики, 3 типа:
  - 1) Прогнозные метрики (работающие на упреждающий анализ). Можно использовать:
    - 1) Фактор сложности проекта (рассматривается, как величина, характеризующая объем всех внешних артефактов проекта, умноженная на коэффициент сложности для каждого из артефактов)
    - 2) Сложность плана проекта (это число взаимосвязей между различными работами в плане графике, отнесенное к общему числу работ) — влияет

на резерв времени, которое нужно заложить в проект для учета сдвигов сроков, кроме того, в проект с большим числом взаимосвязей крайне сложно добавлять новых людей.

- 3) Плотность проекта — отношение суммарной продолжительности всех работ в плане графике при их последовательном выполнении к сумме всего времени для текущего выполнения проекта. Она также является критическим фактором выполнения проекта. Есть критическая сложность, при которой проекты в принципе не выполнимы («захлебнутся» на согласованиях).
- 4) Независимость проекта — отношение числа внутренних зависимостей к сумме внутренних и внешних зависимостей
- 2) Метрики бизнес-процессов (диагностические, ретроспективные). В подавляющем большинстве случаев бизнес-процессы описываются на основе графовых структур различной степени сложности и нотации. Предприятие рассматривается, как многоуровневая конструкция в следующем составе: орг. структура → бизнес-процессы → данные об использовании ресурсов. Для описания схемы орг. структуры используются теоретико-множественные представления (например: тройка — множество филиалов, множество подразделений и множество связей подчиненностей). Собственно, бизнес-процесс рассматривается, как трехуровневая структура:
  - 1) Информационные объекты — идентификаторы объектов + набор его атрибутов
  - 2) Бизнес-операции — пара информационный объект + тип операции, которая над ним выполняется (примеры: создание, присваивание значений, архивирование, уничтожение, регистрация, ознакомление, редактирование, согласование, публикация, передача на исполнение (с назначением маршрутов), контроль за исполнением и привязка к другим информационными объектам).
  - 3) Бизнес-функции — кортеж бизнес-операций (последовательностей бизнес-операций с кодом исполнения)

Тогда, бизнес-процесс — сочетание бизнес-функций.

Визирование — утверждение в своей части. Некоторые комбинации операций для некоторых объектов могут быть запрещены.

Формальная модель бизнес-процесса — граф управления бизнес-функциями. К полученному графу можно полностью применять идеологию управления программным продуктом. Более того, можно использовать автоматизированные методы построения бизнес-процесса (например, с помощью порождающих грамматик). Возможно много вариантов выполнения бизнес-процессов и требуется отбирать предпочтительные варианты:

- 1) Путем формирования критериев отбора (учитывать два обязательных условия: в бизнес-процессе должны быть задействованы все входящие в него функции и требуется учитывать ресурсные ограничения)
- 2) Путем имитационного моделирования (делается у нас на лаб. работах). Сетями Петри проигрываются ветвления, тупики и исключения, средствами GPSS проигрывается бизнес-процесс как система массового обслуживания.

## §4. Метрики для оценки бизнес-процесса как графа

Если бизнес-процесс представлен в виде графовой структуры, то для оценки его качества используются практически те же метрики, что и для графов ПО. В первую очередь: сцепление и связность.

---

## Метрики сцепления бизнес-процесса (традиционные):

- 1) По данным, если:
  - 1) а вызывает b
  - 2) b возвращает управление к а
  - 3) Вся информация передаваемая между а и b представляется значения параметров при вызове
  - 4) Каждый параметр является элементарным информационным объектом)
- 2) По шаблону, если:
  - 1) а вызывает b
  - 2) b возвращает управление к а
  - 3) Вся информация передаваемая между а и b представляется значения параметров при вызове
  - 4) Хотя бы один параметр является составных информационным объектом, то объектом, имеющим внутреннюю структуру),
- 3) По управлению, если:
  - 1) а вызывает b
  - 2) b возвращает управление к а
  - 3) Вся информация передаваемая между а и b представляется значения параметров при вызове
  - 4) Хотя бы один параметр является управляющим объектом  
Управление может пониматься в двух смыслах:
    - 1) Констатирующий объект описывает ситуацию, которая произошла (пример: деньги поступили на счет)
    - 2) Иницирующий объект используется для декларирования определенных действий, в вызываемой бизнес-функции и имеет приказной характер (например: выплаты зарплату за январь)
- 4) По общей области, если они ссылаются на одну и ту же область глобальных данных
- 5) По содержимому, если одна ссылается внутри другой любым способом

Эти характеристики влияют на следующие общие характеристики бизнес-процесса:

- 1) Устойчивость к волновому эффекту (ошибка в одной функции вызывает каскадные ошибки в других бизнес-функциях или процессах)
- 2) Легкость модификации
- 3) Понятность
- 4) Повторная используемость (модульность)

Самое неудачное сцепление — сцепление по содержимому.

Нормальная бизнес-функция означает, что деятельность подразделения абсолютно прозрачна.

---

## Метрики связности бизнес-процесса:

**Связность** — мера прочности соединения функциональных и информационных объектов внутри одной бизнес-функции.

Типы связности:

- 1) Функциональная. Функционально-связанная ф-ия — это функция, содержащая объекты, предназначенные для выполнения одной и только одной задачи (например: расчет з/п).
- 2) Последовательная. Последовательно-связанная ф-ия — функция, объекты которой поставлены каскадно (выходные данные одной задачи служат входными данными для другой).

- 3) Информационная. Информационно-связанная ф-ия — функция, содержащая объекты, использующие одни и те же входные или выходные данные.
- 4) Процедурная. Процедурно-связанная ф-ия — функция, объекты которой включены в различные и возможно не связанные подзадачи, в которых управление переходит от предыдущей задачи к последующей. В последовательно-связанных переходят данные, а здесь — управление. Здесь фиксируется порядок выполнения различных ф-ий.
- 5) Временная. Временно-связанная ф-ия — функция, объекты которой включены в подзадачи, связанные временем исполнения (если какой-то временной сценарий исполнения). Пример: регламент постановки на сигнализацию.
- 6) Логическая. Логически-связанная ф-ия — функция, объекты которой содействуют решению одной общей подзадачи, для которой эти объекты отобраны во внешнем мире. Логически связанные подзадачи должны обладать единым интерфейсом с внешним миром. Семантика (смысл) каждого параметра зависит от используемой задачи.
- 7) Случайная (это плохо, этого делать нельзя)

Традиционные для бизнес процессов нотации (UML, DEF, сети Петри и т. д.) — варианты графовых моделей. Для них характеристики связности и сцепления оцениваются в зависимости от специфики нотации. Но сюжет все-равно сохраняется.

## §4. Возможности анализа бизнес-процессов с использованием сетей Петри

**Сеть Петри** — оргграф с вершинами двух типов: позиции и переходы, причем дугами могут соединяться только вершины различных типов. В позиции сети помещаются специальные маркеры, их перемещение отображает динамику моделируемой системы. Изменение маркировки (движение маркеров) происходит в результате выполнения (срабатывания перехода) на основе соответствующего внешнего события.

Для срабатывания перехода требуется:

- 1) Само внешнее событие
- 2) Наличие маркера во всех его входных позициях

Фактически сеть Петри разделяет системы на активные (переходы) и пассивные (позиции) элементы.

На практике применяются различные, более сложные модификации сетей Петри.

Бывают трех типов:

- 1) Введение иерархии
- 2) Определение различий в маркерах (цветные сети Петри — маркеры разных цветов имеют разные хар-ки)
- 3) Введение многоместных позиций, содержащих несколько маркеров (в таких сетях задается дисциплина поведения маркеров)

Динамическое моделирование — основное применение сетей Петри.

Можно решать задачи с помощью сетей Петри:

- 1) Механизмы взаимодействия процессов (последовательность, параллелизм, альтернативы)
- 2) Временные отношения между выполняемыми процессами (одновременность, наложение, поглощение, сравнение времен запуска завершения и т. д.)
- 3) Абсолютное время (длительность процесса, время запуска)
- 4) Управление исключениями

Идеи сетей Петри могут быть развиты в имитационно-схемном моделировании.

Имитационная схема содержит в себе 3 взаимосвязанных компонента:

- 1) Собственно исполнительная структура — граф, описывающий потенциально возможные взаимодействия процессов, объектов и ресурсов
- 2) Сценарий — определяет ограничения на потенциально возможные последовательности
- 3) Режиссер — управляет исполнительной структурой в соответствии со сценарием

## §5. KPI

**KPI** — ключевые показатели эффективности (метрики для отслеживания бизнес цели).

**KPI** — показатель достижения успеха в определенной деятельности или достижение определенных целей.

Для бизнес процессов (это измерители эффективности, производительности и результативности бизнес процессов), для рабочих процессов (набор метрик, определяемых для оценки эффективности, но трактуемых по другому).

Каждый KPI определяется для конкретного процесса, он может быть докомпозирован на показатели для отдельных сотрудников и ролей, а также агрегирован для подразделения и организации в целом.

Помимо, собственно, метрики определяется её целевое значение (что хочется) и шкала оценки фактического показателя.

---

### Характеристики целей при организации управления по целям

**Суть управления по целям** — система официальных целей на каждом уровне управления дополняется индивидуализированными и персонализированными целями подразделения и сотрудников.

**Управление по целям** — это метод управленческой деятельности, предусматривающий предвидение возможных результатов деятельности и планирования путей их достижения.

На первый взгляд кажется, что разные предприятия имеют разные цели целесообразно разным сферам деятельности, в то же время любое управление предполагает общую методику, то есть возможность:

- Упорядочить управленческую информацию (свести все аспекты управления к небольшому числу основных измерителей)
- Проверять достижения целей на практике
- Прогнозировать поведение сотрудников
- Оценивать целесообразность управленческих решений ещё в процессе их выработки
- Предоставлять возможность анализа прошлого опыта управления

Предложено несколько классификаторов для выделения ключевых сфер деятельности компании, по которым устанавливаются цели.

- Автор Питтер-Друккер — 8 ключевых сфер:
  - Рыночное положение
  - Инновации
  - Производительность
  - Материальные и финансовые ресурсы
  - Прибыльность
  - Работа и обучение управленцев (менеджеров)
  - Работа рядового персонала и отношения с ним



- Ответственность перед обществом
- Сбалансированная система показателей:
  - Финансы
  - Клиенты
  - Процессы
  - Персонал и развитие

На практике при необходимости все классификации дополняются аспектом «Безопасность» и «Государство»

Аспекты классификации:

- На какой промежуток времени требуется устанавливать цели (при постановке цели требуется найти равновесие между ближайшим будущим и долгосрочными планами)
- Для определения периода постановки целей рекомендуется использовать тот же период, что и для годового бюджетного планирования. Сроки эти специфичны и зависят от основного бизнес-процесса (например, финансовый год — 1 января - 31 декабря, учебный год — 1 сентября - 30 июня)
- Постановка целей и выбор KPI должны быть регламентированы специальным документом. Для управления этим процессом выбирается комплект KPI, в котором должны быть хорошо представлены все заинтересованные лица. Результатом должен быть согласованный список показателей, по которым происходит оценка работы на следующий год.

---

## Требования KPI и примеры их реализации

- 1) **Адресная принадлежность.** каждый KPI закрепляется за конкретным сотрудником или группой, которые несут ответственность за соответствующий результат
- 2) **Правильная ориентация.** KPI должны быть привязаны к корпоративным стратегическим целям, ключевым бизнес-процессам и проектам развития.
- 3) **Достижимость.** Утвержденные показатели и нормативы должны быть достижимы. Достижение цели должно быть связано с приложением значительных усилий, но в то же время вероятность ее достижения должна быть не менее 70 – 80%.
- 4) **Открытость к действиям.** Значения KPI рассчитываются на основе актуальных данных, то есть пользователи должны иметь возможность вмешиваться в процессы, чтобы улучшить результаты работы, пока время еще не упущено.
- 5) **Обеспечение прогнозирования.** KPI количественно оценивают факторы, влияющие на стоимость бизнеса, то есть они являются показателями, определяющими желательные будущие результаты.
- 6) **Ограниченность.** KPI должны фокусировать усилия исполнителей на достижении нескольких высокоприоритетных задач, а не рассеивать их на слишком многие предметы.
- 7) **Легкость восприятия.** KPI должны быть легкими для понимания.
- 8) **Сбалансированность и взаимосвязанность.** KPI должны быть сбалансированы и «поддерживать» друг друга, а не конфликтовать друг с другом.
- 9) **Инициирование изменений.** Измерения KPI должны вызывать в организации цепную реакцию положительных изменений, особенно если за процессом следит руководство компании. Вместе с предыдущим требованием должны отслеживаться руководством.
- 10) **Простота измерения.** KPI должны быть понятны и доступны для измерения пользователям.
- 11) **Подкрепленность соответствующими индивидуальными стимулами.** Показатели должны способствовать мотивации пользователей.
- 12) **Релевантность.** Воздействие KPI со временем ослабевает, поэтому их следует периодически пересматривать и «освежать».
- 13) **Сопоставимость.** KPI должны быть сопоставимыми, чтобы одни и те же показатели можно было сравнить в двух подобных ситуациях. Например, нельзя сравнивать значения такого показателя, как средний чек (KPI – отношение среднедневной

выручки к количеству чеков за день), для магазинов одного формата, но расположенных в областном центре и «в глубинке».

KPI должны быть объединены в единый критерий.

## Организационные моменты

Если лабы сданы до зачетных недель + рубежки написаны вовремя => 91 балл.

Если больше 40, но меньше 60 — досдавать лабы можно будет только до зачетной недели, если не успели — сдаем экзамен. На зачетной неделе будет одна консультация, на которой можно будет только написаны рубежки.

Рубежку можно написать либо на лекции 18-го, либо 20-го.