



Модели качества программного обеспечения и методы его оценивания

Доктор технических наук, профессор Соколов Б.В.

*С.-Петербургский институт информатики и автоматизации РАН,
С.-Петербург,
14 линия ВО, 39, СПИИ РАН,*

Содержание

- 1. Основные понятия и определения.*
- 2. Модели качества программного обеспечения*
- 3. Программный код и его метрики*
- 4. Методы оценивания качества программного обеспечения*



Основные понятия и определения

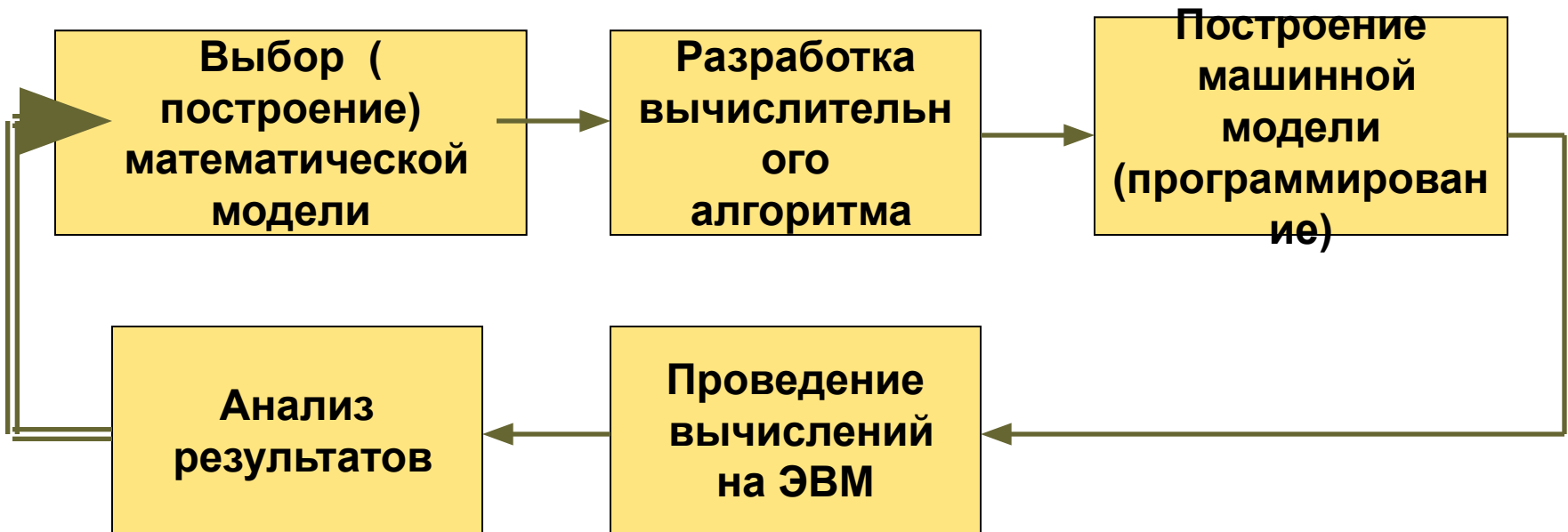
Основные понятия и определения

Качество – совокупность характеристик объекта, определяющих его способности удовлетворять установленным или предполагаемым потребностям (Международный стандарт ISO-8402).

Качествоведение – отрасль знаний, в которой изучаются закономерности получения и обработки информации о качестве объекта на всех этапах его жизненного цикла.

Квалиметрия – раздел качествоведения, в котором разрабатываются методологические и методические основы количественного оценивания качества продукции, средства обеспечения единства форм оценивания указанного качества и достижение требуемой точности

Основные понятия и определения



Качество модели – совокупность свойств модели, обуславливающих ее пригодность для использования по назначению.

Возможные предназначения модели – имитация функционирования объекта-оригинала в целях более глубокого познания его свойств, оптимизации его характеристик, прогнозирования его поведения, принятие управленческих решений

Основные понятия и определения

Прямые задачи – анализ качества продукции

Обратные задачи – синтез продукции с заданными (требуемыми) свойствами, управление качеством продукции с целью придания ей необходимых свойств.

В квалиметрии моделей главную роль играют обратные задачи, т.к. модели являются основным предметом разработки.

Основные понятия и определения.

Качество системы определяемая степенью удовлетворения системой заявленных и подразумеваемых потребностей различных заинтересованных сторон, которая позволяет, таким образом, оценить достоинства.

Эти заявленные и подразумеваемые потребности представлены в международных стандартах серии **SQuaRE (Требования и оценка качества систем и программного обеспечения)** посредством **моделей качества**, которые представляют качество продукта в виде разбивки на классы характеристик, которые в отдельных случаях далее разделяются на подхарактеристики. (Некоторые подхарактеристики разделяются далее на под-подхарактеристики.) Подобная **иерархическая декомпозиция** обеспечивает удобную разбивку качества продукта на классы. Однако множество подхарактеристик, связанных с характеристикой, избранной для представления типичных проблем, необязательно будет исчерпывающим.

Измеримые, связанные с качеством свойства системы называют **свойствами качества**, связанными с соответствующими показателями качества. Чтобы прийти к показателям характеристики или подхарактеристики качества в случаях, когда характеристика или подхарактеристика не может быть непосредственно измерена, необходимо идентифицировать подмножество свойств, которое в совокупности покрывает характеристику или подхарактеристику, получить показатели качества для каждого свойства и, объединив их в вычислительном отношении, достигнуть полученного показателя качества, соответствующего характеристике или подхарактеристике качества.

Структура, используемая для моделей качества



Модель качества продукта



Модель качества при использовании



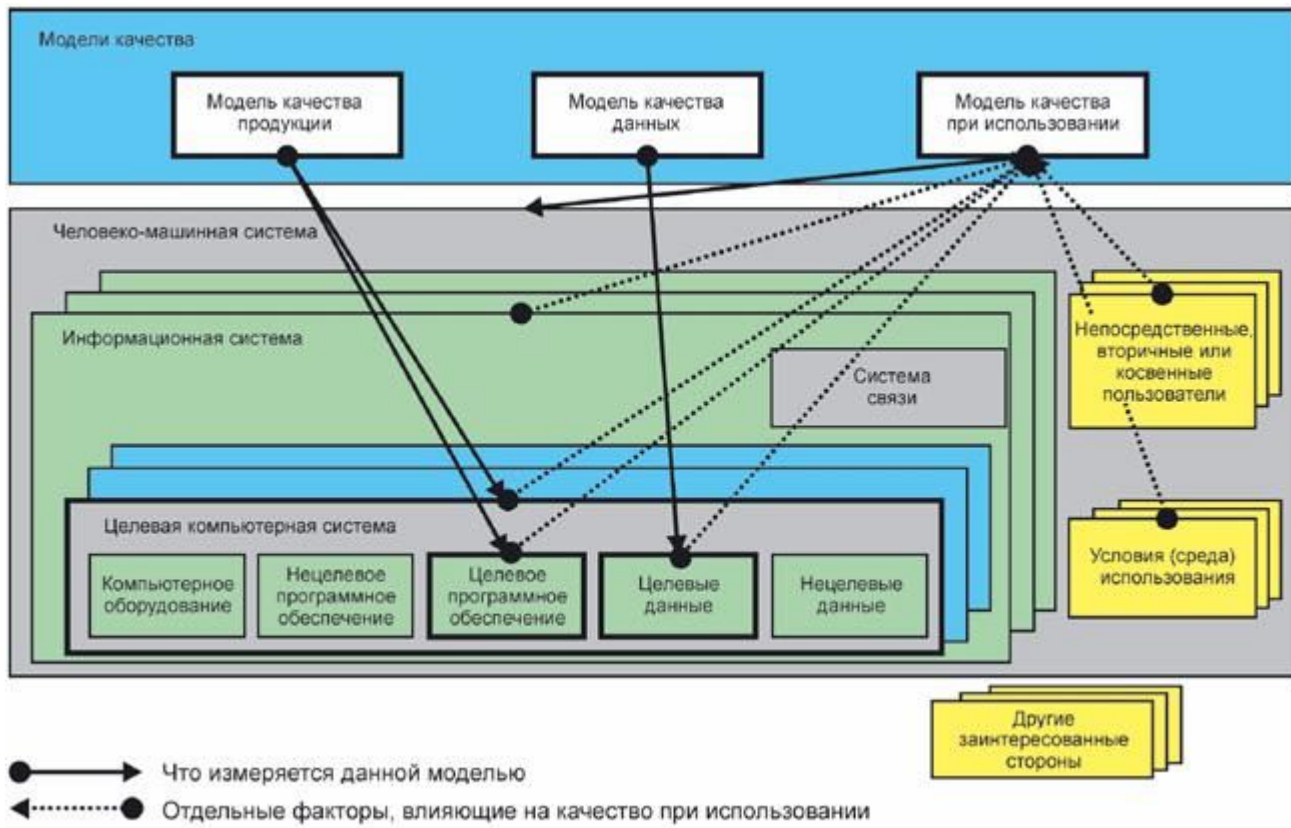
Цели моделей качества


Целью модели качества продукта является компьютерная система, в которую входит целевой программный продукт, а **цель модели качества при использовании** - это совокупная человеко-машинная система, которая включает в себя и целевую компьютерную систему, и целевой программный продукт.

В целевую компьютерную систему входят также компьютерное оборудование, нецелевые программные продукты, нецелевые данные и целевые данные, которые, в свою очередь, являются объектом анализа модели качества данных. Целевая компьютерная система является частью информационной системы, в состав которой могут быть также включены одна или более компьютерных систем и системы связи, такие как локальная сеть и Интернет. В состав информационной системы в более крупной человеко-машинной системе (такой как корпоративная система, встроенная система или крупномасштабная система управления) могут входить пользователи, техническая и физическая среда использования. Рамки целевой системы определяются исходя из области применения требований или оценки и из того, кто рассматривается в качестве пользователей.

***Пример** - Если в качестве пользователей самолета с компьютерной системой управления полетом рассматривать пассажиров, то система, от которой они зависят, включает летный экипаж, сам самолет, аппаратное и программное обеспечение системы управления полетом. В случае, если в качестве пользователей рассматривать летный экипаж, то система, от которой они зависят, состоит только из самого самолета и системы управления полетом.*

Цели моделей качества





Модели качества программного обеспечения

Руководящие документы

ГОСТ Р ИСО/МЭК 25010-2015 Информационные технологии (ИТ). Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов

ГОСТ Р ИСО/МЭК 25010-2015

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационные технологии

СИСТЕМНАЯ И ПРОГРАММНАЯ ИНЖЕНЕРИЯ

Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов

Information technology. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models

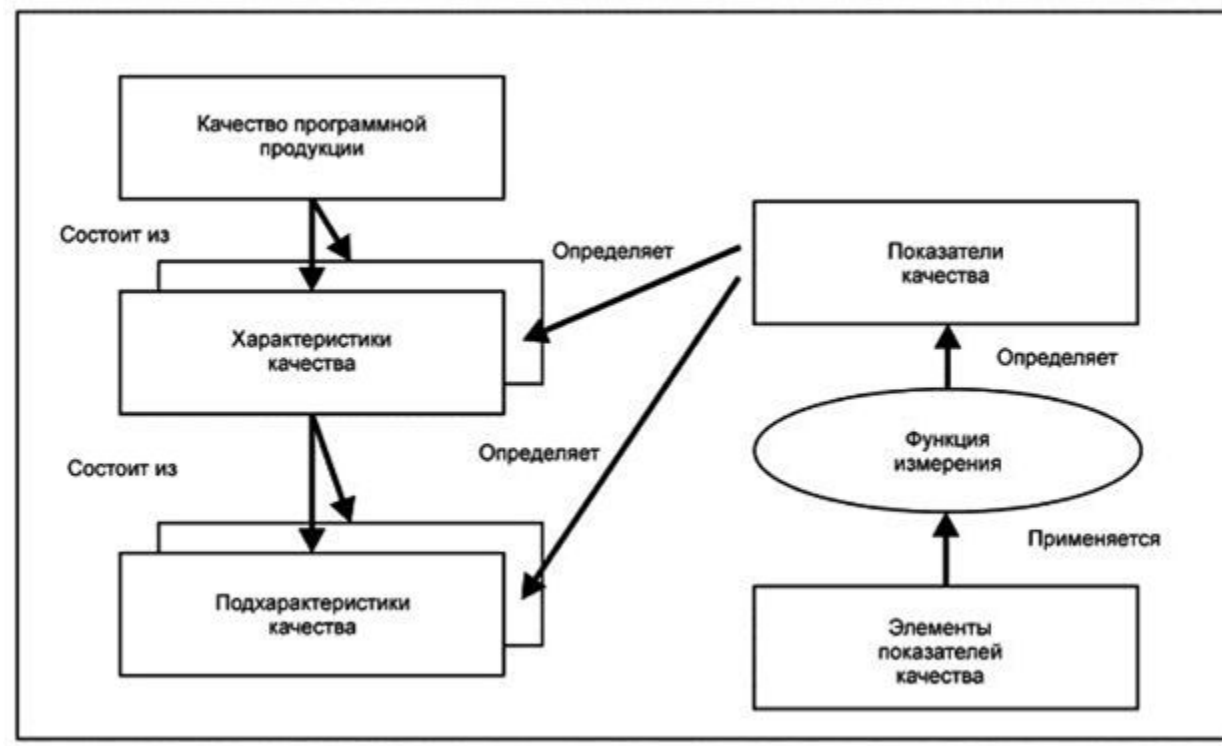
Структура, используемая для моделей качества

Свойства качества - это неотъемлемые свойства программного обеспечения, которые обеспечивают качество. Свойства качества могут быть разделены на одну или несколько подхарактеристик.

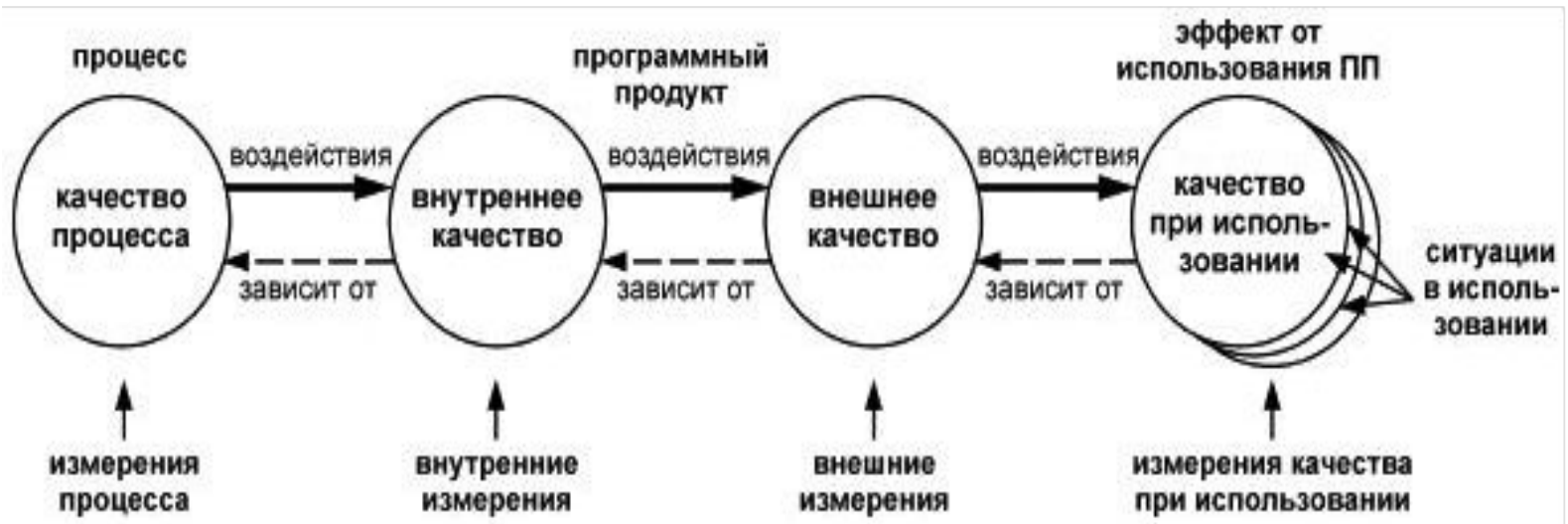
Измеряются свойства качества посредством **метода измерения**.

Метод измерения представляет собой логическую последовательность операций, используемых для количественного определения свойств относительно конкретной шкалы. Результат применения метода измерения **называют элементом показателя качества (ЭПК)**. Характеристики и подхарактеристики качества могут быть количественно определены с помощью функции измерения. Функция измерения - это алгоритм, используемый для объединения элементов показателя качества. Результат применения функции измерения называют показателем качества программного обеспечения. Таким образом показатели качества программного обеспечения становятся количественными показателями характеристик и подхарактеристик качества. Для измерения характеристики или подхарактеристики качества могут быть использованы несколько показателей качества программного обеспечения.

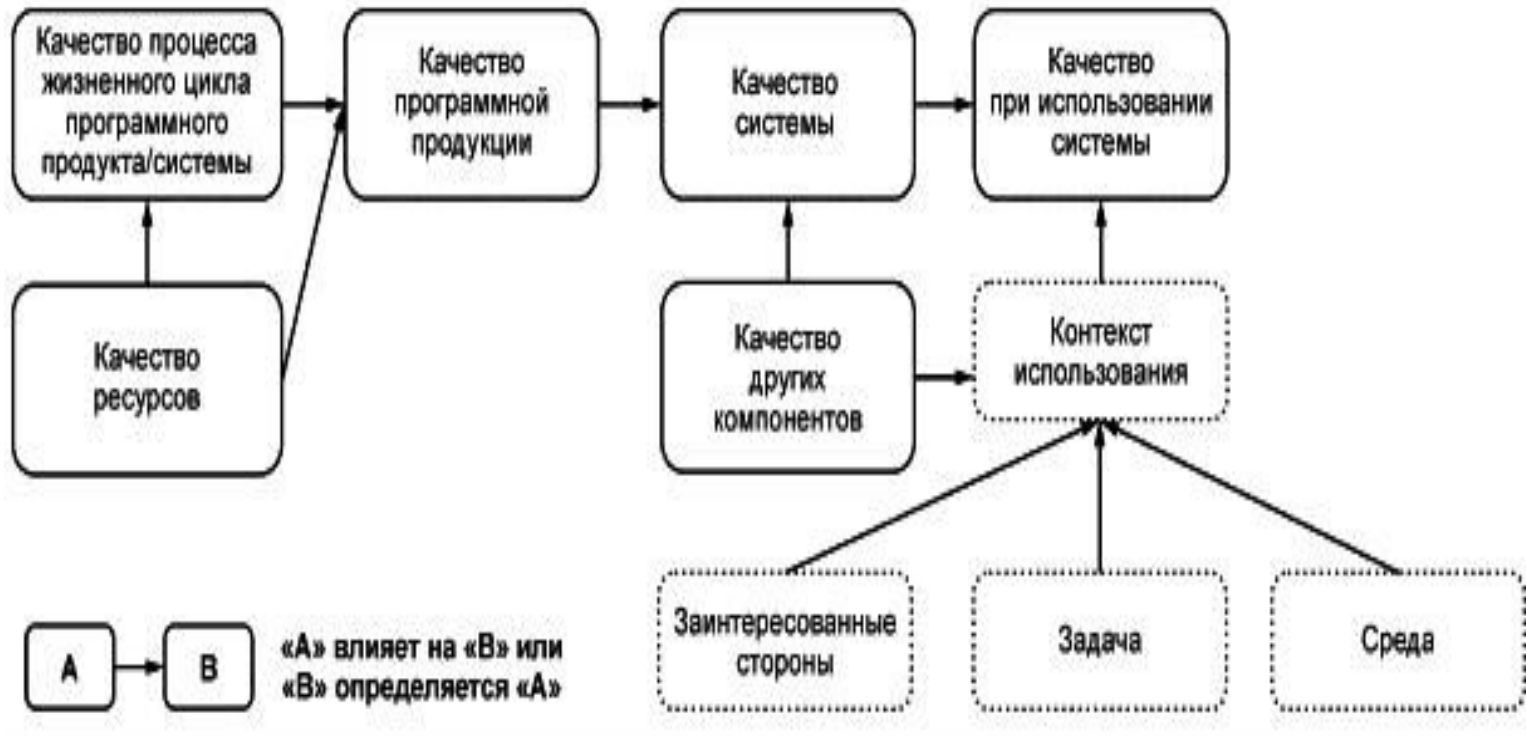
Эталонная модель измерения качества программного продукта



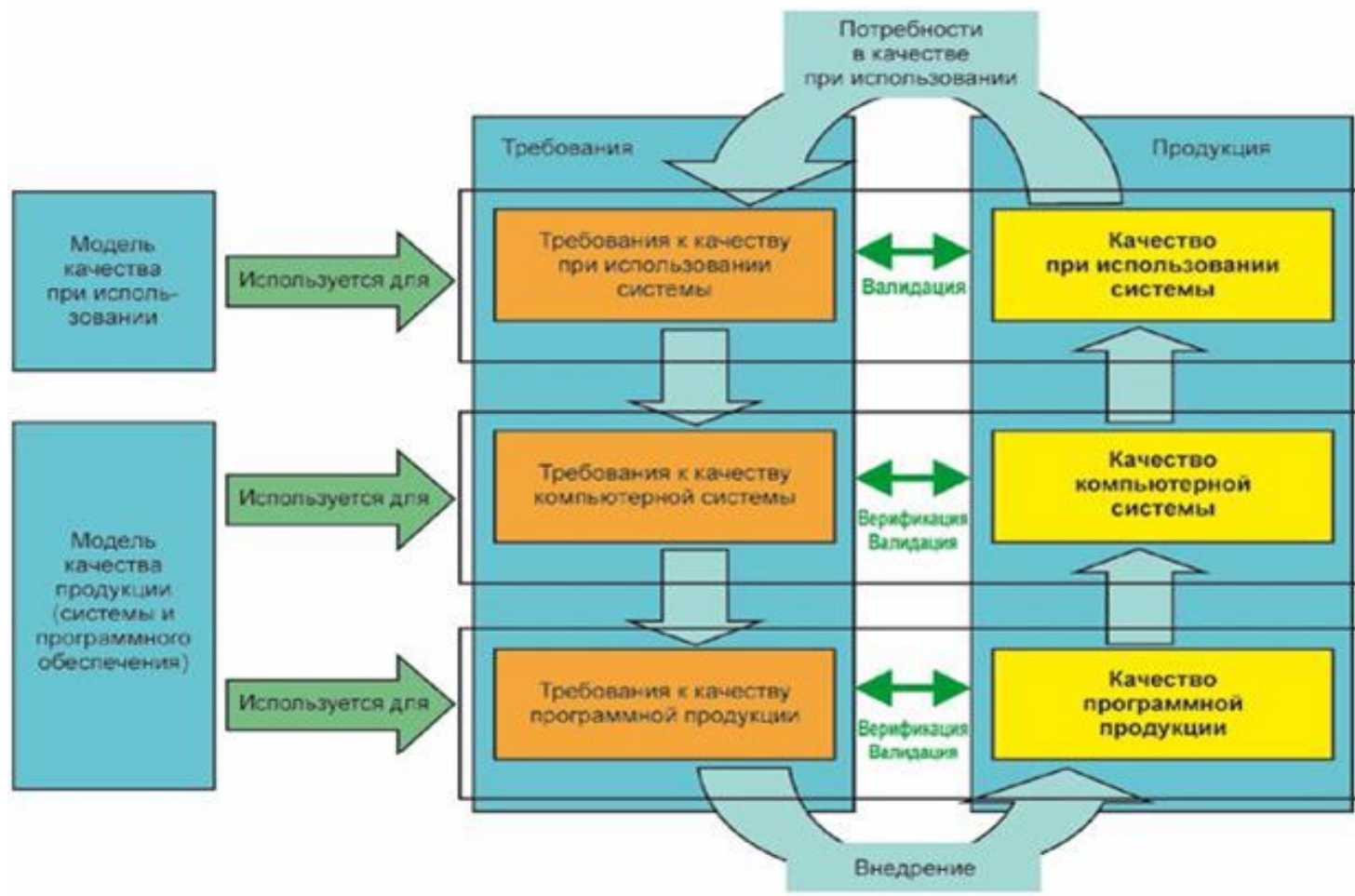
Качество в жизненном цикле



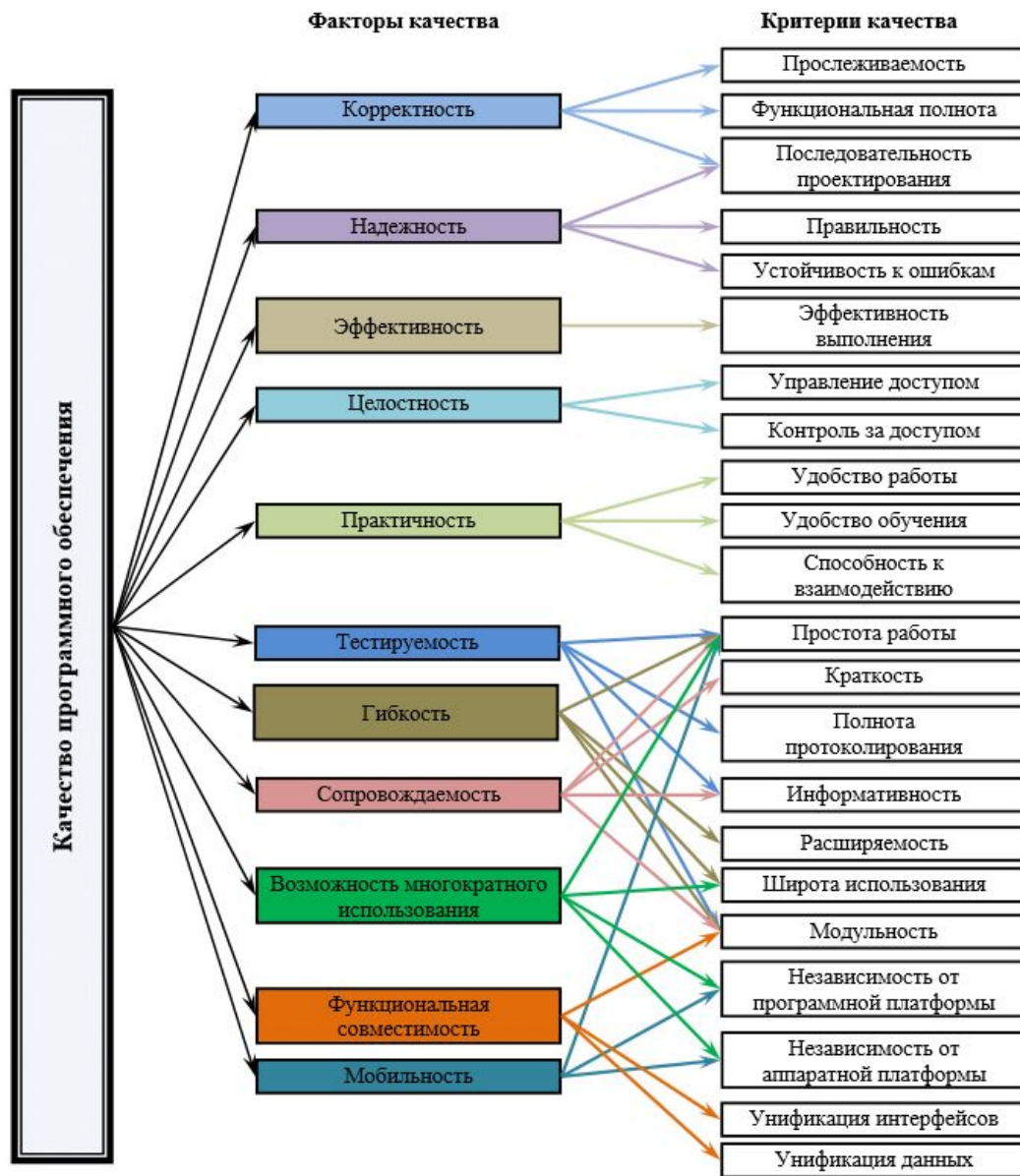
Целевые объекты модели качества и их взаимосвязь



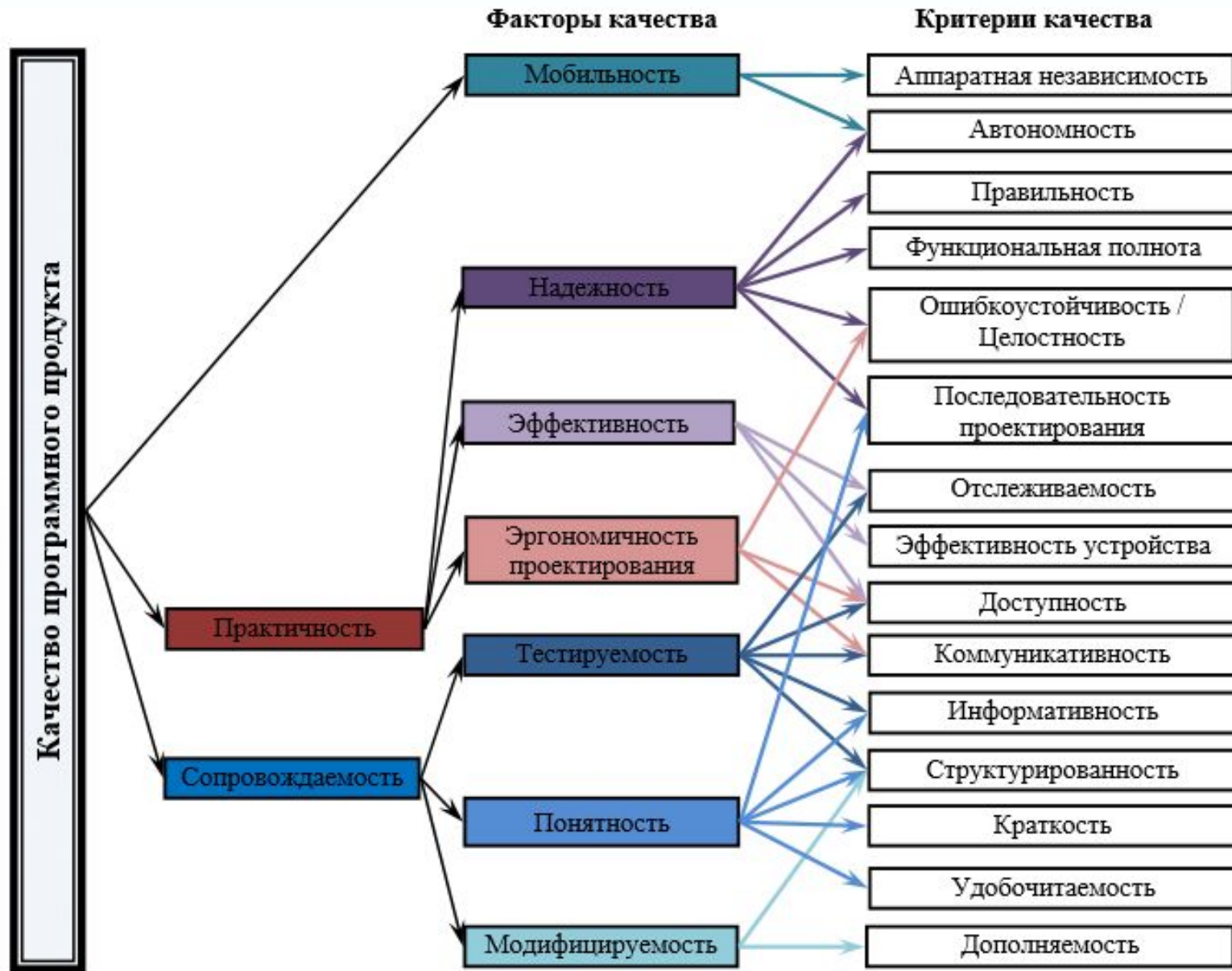
Модель жизненного цикла качества системы/программного обеспечения



Модель качества программного обеспечения МакКола



Модель качества программного продукта Бозма



Модель качества программного обеспечения FURPS/FURPS+

Модель FURPS предложена Грейди и Hewlett Packard.

Акроним FURPS, используемый в обозначении модели, обозначает следующие категории требований к качеству ПО:

- **F**unctionality (Функциональность) /особенности, возможности, безопасность/;
- **U**sability (Практичность) /человеческий фактор, эргономичность, пользовательская документация/;
- **R**eliability (Надежность) /частота отказов, восстановление информации, прогнозируемость/;
- **P**erformance (Производительность) /время отклика, пропускная способность, точность, доступность, использование ресурсов/;
- **S**upportability (Эксплуатационная пригодность) /тестируемость, расширяемость, адаптируемость, сопровождаемость, совместимость, конфигурируемость, обслуживаемость, требования к установке, локализуемость/.
- Символ «+» расширяет FURPS модель, добавляя к ней:
 - ограничения проекта (ограничения по ресурсам, требования к языкам и средствам разработки, требования к аппаратному обеспечению);
 - интерфейс (ограничения накладываемые на взаимодействие с внешними системами);
 - требования к выполнению,
 - физические требования,
 - требования к лицензированию.

Модель качества программного обеспечения Гецци

Карло Гецци и соавторы различают качество продукта и процесса.

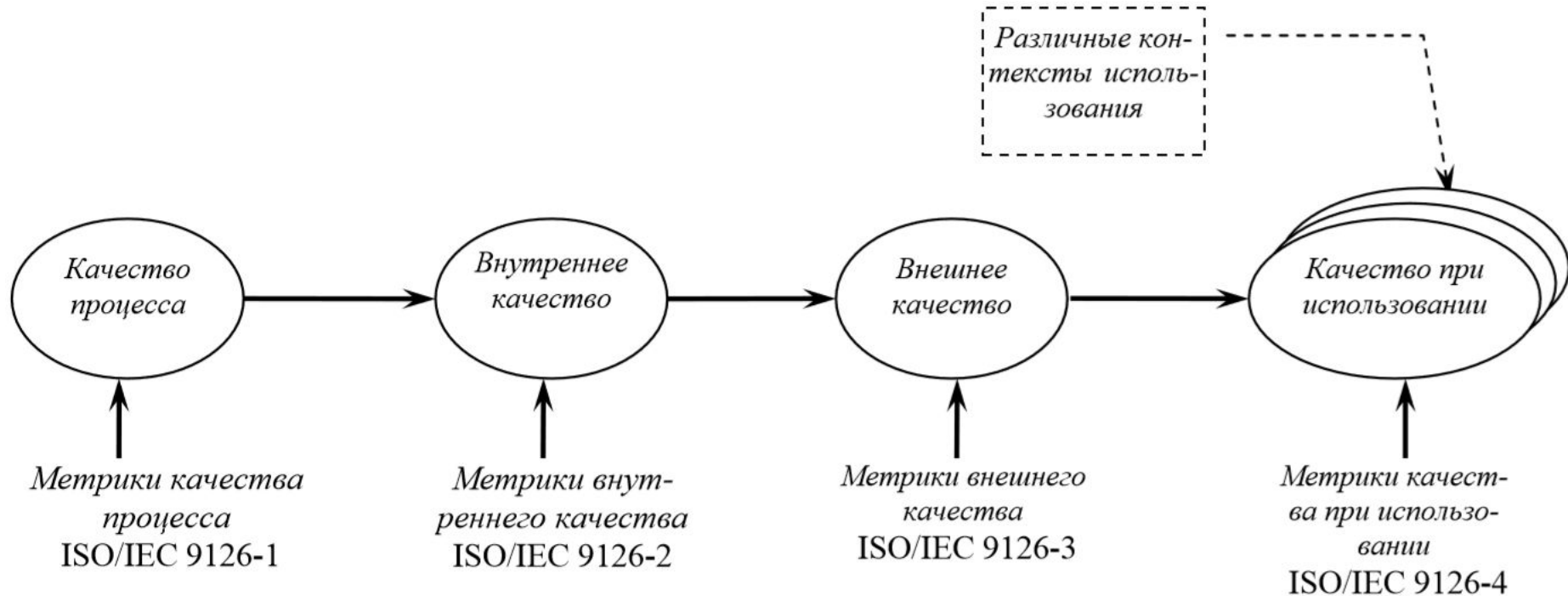
Характеристики ПО Гецци:

- целостность,
- надежность и устойчивость,
- производительность,
- практичность,
- верифицируемость,
- сопровождаемость,
- возможность многократного использования,
- мобильность,
- понятность,
- возможность взаимодействия,
- эффективность,
- своевременность реагирования,
- видимость процесса разработки

Некоторые другие модели качества программного обеспечения

- **Модель качества Дроми**
- **Модель качества SATC**
- **Модель качества ISO 9126**
- **Модель качества QMOOD**
- **Модель качества Хосрави**
- **Модель качества Шармоа**

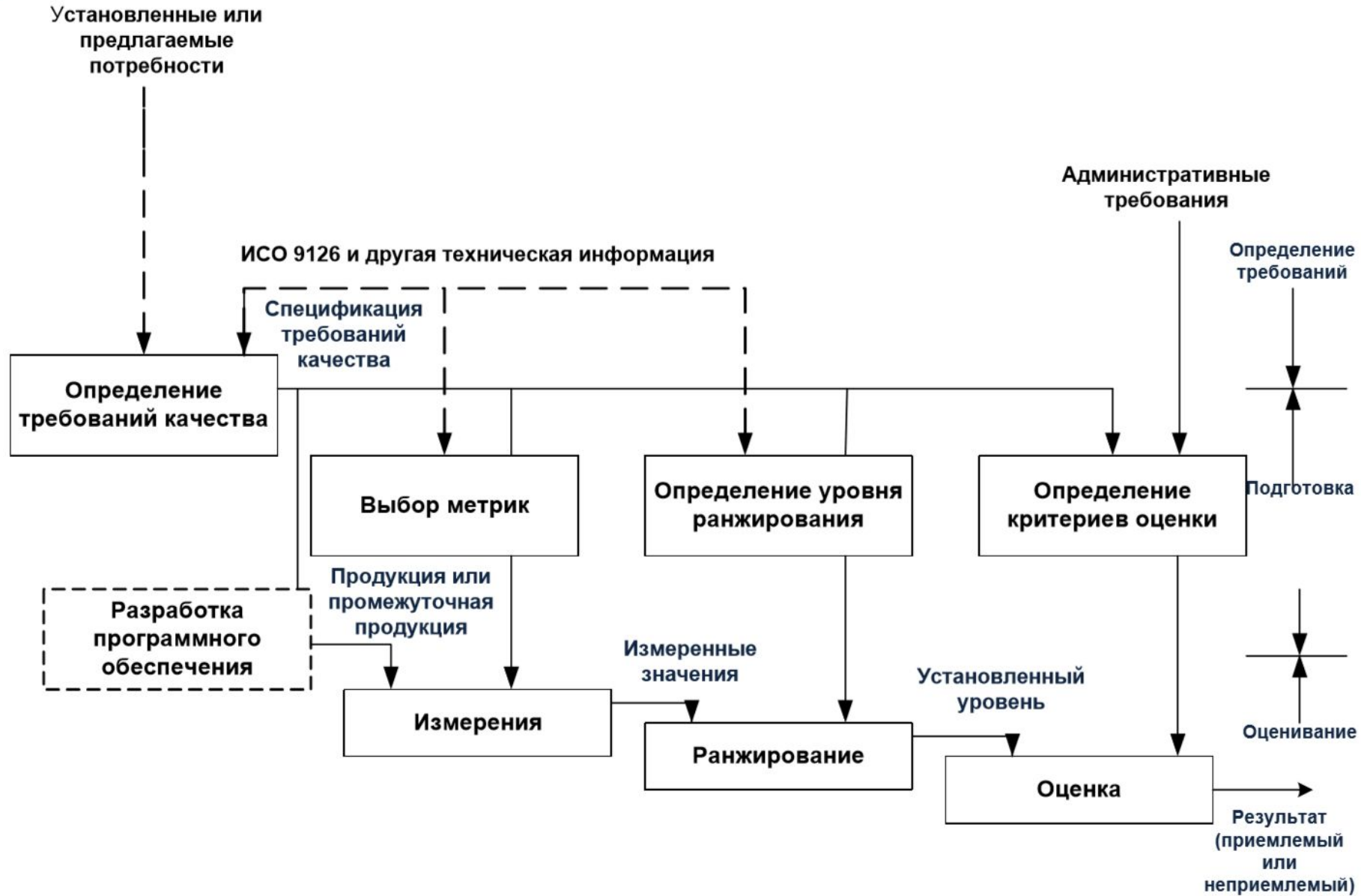
Основные аспекты качества программного обеспечения



Факторы и атрибуты внешнего и внутреннего качества программного обеспечения



Факторы и атрибуты внешнего и внутреннего качества программного обеспечения



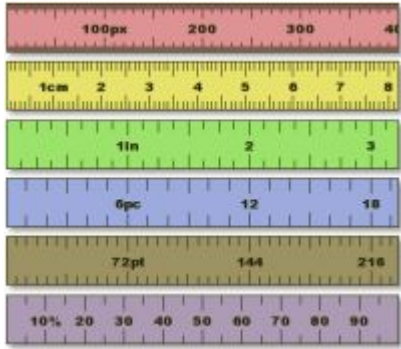
Сравнительный анализ моделей качества программного обеспечения

Характеристики качества	МакКолл [4-6]	Бозм [7]	FURPS/ FURPS+ [8]	Геци [9]	Дроми [10]	ISO 9126 [2]	Казман [16]	Хосрави [18]	Шармоа [19]
Корректность				+					
Надежность	+	+	+	+	+	+	+		+
Корректность	+								
Эффективность	+	+	+	+	+	+	+		+
Гибкость	+			+			+	+	
Функциональность			+		+	+	+		+
Эргономичность проектирования		+							
Целостность				+					
Функциональная совместимость	+								
Сопровождаемость	+	+	+	+	+	+	+		+
Модифицируемость		+							
Производительность			+						
Мобильность	+	+		+	+	+			+
Зрелость процесса					+				
Возможность многократного использования	+			+				+	
Устойчивость								+	
Масштабируемость								+	
Безопасность			+				+		
Эксплуатационная пригодность			+						
Тестируемость	+	+					+		
Понятность		+	+						
Практичность	+		+	+	+	+	+	+	+



Программный код и его метрики

Программный код и его метрики



Одной из тем в программировании, к которым интерес периодически то появляется, то пропадает, является вопрос метрик кода программного обеспечения. В крупных программных средах время от времени появляются механизмы подсчета различных метрик. Волнообразный интерес к теме так выглядит потому, что до сих пор в метриках не придумано главного — что с ними делать. То есть даже если какой-то инструмент позволяет хорошо подсчитать некоторые метрики, то что с этим делать дальше зачастую непонятно. Конечно, метрики — это и контроль качества кода (не пишем большие и сложные функции), и «производительность» (в кавычках) программистов, и скорость развития проекта.

Программный код и его метрики

Количественные метрики

Прежде всего, следует рассмотреть количественные характеристики исходного кода программ (в виду их простоты). **Самой элементарной метрикой является количество строк кода (SLOC)**. Данная метрика была изначально разработана для оценки трудозатрат по проекту. Однако из-за того, что одна и та же функциональность может быть разбита на несколько строк или записана в одну строку, метрика стала практически неприменимой с появлением языков, в которых в одну строку может быть записано больше одной команды. Поэтому **различают логические и физические строки кода**. Логические строки кода — это количество команд программы. Данный вариант описания так же имеет свои недостатки, так как сильно зависит от используемого языка программирования и стиля программирования.

Кроме SLOC к количественным характеристикам относят также:

количество пустых строк,

количество комментариев,

процент комментариев (отношение числа строк, содержащих комментарии к общему количеству строк, выраженное в процентах),

среднее число строк для функций (классов, файлов),

среднее число строк, содержащих исходный код для функций (классов, файлов),

среднее число строк для модулей.

Программный код и его метрики

Метрики Холстеда. Данные метрики основаны на следующих показателях:

n1 — число уникальных операторов программы, включая символы-разделители, имена процедур и знаки операций (словарь операторов),

n2 — число уникальных операндов программы (словарь операндов),

N1 — общее число операторов в программе,

N2 — общее число операндов в программе,

n1' — теоретическое число уникальных операторов,

n2' — теоретическое число уникальных операндов.

Учитывая введенные обозначения, можно определить:

$n = n1 + n2$ — словарь программы,

$N = N1 + N2$ — длина программы,

$n' = n1' + n2'$ — теоретический словарь программы,

$N' = n1 \cdot \log_2(n1) + n2 \cdot \log_2(n2)$ — теоретическая длина программы (для стилистически корректных программ отклонение N от N' не превышает 10%)

Программный код и его метрики

$V=N \cdot \log_2 n$ — объем программы,

$V'=N' \cdot \log_2 n'$ — теоретический объем программы, где n' — теоретический словарь программы.

$L=V'/V$ — уровень качества программирования, для идеальной программы $L=1$

$L'=(2n_2)/(n_1 \cdot N_2)$ — уровень качества программирования, основанный лишь на параметрах реальной программы без учета теоретических параметров,

$E_c=V/(L')^2$ — сложность понимания программы,

$D=1/L'$ — трудоемкость кодирования программы,

$y' = V/D^2$ — уровень языка выражения

$I=V/D$ — информационное содержание программы, данная характеристика позволяет определить умственные затраты на создание программы

$E=N' \cdot \log_2(n/L)$ — оценка необходимых интеллектуальных усилий при разработке программы, характеризующая число требуемых элементарных решений при написании программы

При применении **метрик Холстеда** частично компенсируются недостатки, связанные с возможностью записи одной и той же функциональности разным количеством строк и операторов.

Еще одним типом метрик ПО, относящихся к количественным, являются **метрики Джилба**. Они показывают сложность программного обеспечения на основе насыщенности программы условными операторами или операторами цикла. Данная метрика, не смотря на свою простоту, довольно хорошо отражает сложность написания и понимания программы, а при добавлении такого показателя, как максимальный уровень вложенности условных и циклических операторов, эффективность данной метрики значительно возрастает.

Программный код и его метрики

- *Метрики сложности потока управления программы*
- *Метрики сложности потока управления данными*
- *Метрики сложности потока управления и данных программы*
- *Объектно-ориентированные метрики*
- *Гибридные метрики*

Вывод: в настоящее время ни одной универсальной метрики не существует. Любые контролируемые метрические характеристики программы должны контролироваться либо в зависимости друг от друга, либо в зависимости от конкретной задачи, кроме того, можно применять гибридные меры, однако они так же зависят от более простых метрик и также не могут быть универсальными. Строго говоря, любая метрика — это лишь показатель, который сильно зависит от языка и стиля программирования, поэтому ни одну меру нельзя возводить в абсолют и принимать какие-либо решения, основываясь только на ней.

Общие показатели качества программных средств

Характеристики качества	Мера
Надежность	
<i>Завершенность:</i>	
<ul style="list-style-type: none"> ■ наработка на отказ при отсутствии рестарта 	Часы
<i>Устойчивость:</i>	
<ul style="list-style-type: none"> ■ наработка на отказ при наличии автоматического рестарта; ■ относительные ресурсы на обеспечение надежности и рестарта 	Часы %
<i>Восстанавливаемость:</i>	
<ul style="list-style-type: none"> ■ длительность восстановления 	Минуты
<i>Доступность-готовность:</i>	
<ul style="list-style-type: none"> ■ относительное время работоспособного функционирования 	Вероятность
Эффективность	
<i>Временная эффективность:</i>	
<ul style="list-style-type: none"> ■ время отклика-получения результатов на типовое задание; ■ пропускная способность – число типовых заданий, исполняемых в единицу времени 	Секунды Число в минуту
<i>Используемость ресурсов:</i>	
<ul style="list-style-type: none"> ■ относительная величина использования ресурсов ЭВМ при нормальном функционировании программного средства. 	Вероятность

Частные показатели качества программных средств

Характеристики качества	Мера
<p>Практичность</p> <p><i>Понятность:</i></p> <ul style="list-style-type: none"> ■ четкость концепции ПС; ■ демонстрационные возможности; ■ наглядность и полнота документации. <p><i>Простота использования:</i></p> <ul style="list-style-type: none"> ■ простота управления функциями; автоматического рестарта; ■ комфортность эксплуатации; ■ среднее время ввода заданий; ■ среднее время отклика на задание. <p><i>Изучаемость:</i></p> <ul style="list-style-type: none"> ■ трудоемкость изучения применения ПС; ■ продолжительность изучения; ■ объем эксплуатационной документации; ■ объем электронных учебников. 	<p>Порядковая (отл., хор., удов.,неуд.)</p> <p>Порядковая</p> <p>Секунды Секунды</p> <p>Чел.-часы Часы Страницы Кбайты</p>
<p>Сопровождаемость</p> <p><i>Анализируемость:</i></p> <ul style="list-style-type: none"> ■ стройность архитектуры программ; ■ унифицированность интерфейсов; ■ полнота и корректность документации. 	<p>Порядковая</p>

Частные показатели качества программных средств

Характеристики качества	Мера
<p>Изменяемость:</p> <ul style="list-style-type: none">▪ трудоемкость подготовки изменений;▪ длительность подготовки изменений. <p>Стабильность:</p> <ul style="list-style-type: none">▪ устойчивость к негативным проявлениям при изменениях. <p>Тестируемость:</p> <ul style="list-style-type: none">▪ трудоемкость тестирования изменений;▪ длительность тестирования изменений. <p>Мобильность</p> <p>Адаптируемость:</p> <ul style="list-style-type: none">▪ трудоемкость адаптации;▪ длительность адаптации. <p>Простота установки:</p> <ul style="list-style-type: none">▪ трудоемкость инсталляции;▪ длительность инсталляции.	<p>Чел.-часы Часы</p> <p>Порядковая</p> <p>Чел.-часы Часы</p> <p>Чел.-часы Часы</p> <p>Чел.-часы Часы</p>



Методы оценивания качества программного обеспечения

Методы оценивания качества программного обеспечения

Характерные особенности задач многокритериального выбора

Реальные задачи выбора, возникающие на практике, чрезвычайно разнообразны, но всех их объединяет общая схема поиска решения, суть которой состоит в формировании совокупности операций (процедур), проводимых на множестве допустимых альтернатив, в результате чего выделяется множество наилучших (оптимальных) альтернатив.

Для поиска указанных альтернатив в задачах выбора необходимо задать соответствующие критерии (греч. *Kriterion* - мерило для оценки), под которыми в дальнейшем понимаются и правила (признаки), позволяющие сопоставлять и сравнивать допустимые альтернативы друг с другом для выбора наилучшей из них. При этом оценивание альтернатив в сложных инженерно-технических задачах, как правило, осуществляется с использованием нескольких критериальных функций. Данные функции в научно-технической литературе часто называют еще целевыми функциями, показателями качества, показателями эффективности.

Можно указать на 4 основных вида задач выбора, при решении которых необходимо использовать многокритериальный подход. Перечислим указанные виды задач многокритериального выбора:

Характерные особенности задач многокритериального выбора

1-й вид задач, в которых окончательное решение, определяет порядок совместных действий нескольких объектов, эффективность функционирования каждого из которых оценивается отдельными критериальными функциями (например, совместная деятельность СТС при выполнении общей задачи);

2-й вид задач, в которых качество принимаемого решения необходимо оценивать для нескольких вариантов условий воздействия среды на СТС и для каждого варианта вводится отдельная оценка;

3-й вид задач, в которых принятие решения осуществляется поэтапно с использованием на каждом этапе своих критериальных функций (например, оценка эффективности жизненного цикла СТС);

4-й вид задач, в которых качество решения необходимо оценивать с нескольких точек зрения - по отдельным компонентам качества. Например, оценка качества выполнения плана работы системы обслуживания активных подвижных объектов (АПО) может характеризоваться временем окончания всех операций взаимодействия на интервале планирования, количеством израсходованных ресурсов системы обслуживания (СО), объемом выполненных операций.

Характерные особенности задач многокритериального выбора

Анализ показывает, что большинство задач выбора, возникающих на практике, принадлежит к одному из перечисленных выше видов задач или является их комбинацией. Таким образом, при создании, исследовании, применении и развитии сложных технических, экономических, организационных, военно-технических систем оценивание качества соответствующих процессов становится возможным только при использовании нескольких показателей (нескольких целевых, критериальных функций). Это приводит, в свою очередь, к появлению в задачах выбора критериальной неопределенности. Рассмотрим пример, иллюстрирующий причины появления указанной критериальной неопределенности при решении задач выбора на практике.

Характерные особенности задач многокритериального выбора

Следует подчеркнуть, что подобного рода примеров можно привести очень много (далее в данной и последующих главах будут приводиться еще ряд примеров постановки задач многокритериального выбора). Даже в обыденной жизни каждый человек при определении места работы или отпуска испытывает затруднения, связанные с наличием нескольких противоречивых критериев, на основании которых нужно принять окончательное решение.

Одна из **главных особенностей** задач многокритериального выбора состоит в том, что данные задачи не являются корректными в рамках аксиоматики, принятой в классической теории оптимизации и принятия решения. В самом деле, если взять условия примера 4.1, то формальная постановка задачи многокритериального выбора сводится к следующему. Пусть вектор $\bar{x} = \|x_1, x_2, \dots, x_n\|^T$ характеризует основные параметры проектируемого самолета, возможные значения которых задаются множеством допустимых альтернатив $\Delta_{s\beta}$. Качество проектирования самолета оценивается m -скалярными критериальными функциями $f_1(\bar{x}), f_2(\bar{x}), \dots, f_m(\bar{x})$, содержательная интерпретация которых приводилась выше (см. условия примера 4.1). Образует из данных функций вектор $\bar{f}(x) = \|f_1(\bar{x}), f_2(\bar{x}), \dots, f_m(\bar{x})\|^T$.

4.1.1. Характерные особенности задач многокритериального выбора

В указанных условиях задача многокритериального выбора сводится к поиску такого вектора \bar{x}^* , при котором

$$f(\bar{x}) \rightarrow \underset{x \in \Delta_{s\beta}}{extr} \quad (4.1)$$

или по-другому

$$f_1(\bar{x}) \rightarrow \underset{x \in \Delta_{s\beta}}{extr}; f_2(\bar{x}) \rightarrow \underset{x \in \Delta_{s\beta}}{extr}; \dots; f_m(\bar{x}) \rightarrow \underset{x \in \Delta_{s\beta}}{extr} \quad (4.2)$$

Условие существования решения (4.1) или (4.2) может быть записано как условие совпадения решения m -частных задач поиска экстремума по каждому j -му показателю качества на множестве $\Delta_{s\beta}$:

$$x_1^* = x_2^* = x_3^* = \dots = x_m^* \quad (4.3)$$

где $\bar{x}_i^* = \arg \underset{x \in \Delta_{s\beta}}{extr} f_i(\bar{x}), i = 1, \dots, m$

Выполнение условия (4.3) возможно лишь в случае непротиворечивости частных показателей качества проектирования самолета. Однако, как показывает содержательный анализ примера 4.1, указанные показатели являются сугубо противоречивыми и оптимизация параметров проектирования самолета по каждому из них приводит к альтернативным (несовпадающим) решениям.

Характерные особенности задач многокритериального выбора

Таким образом, постановка задачи (4.1) является не корректной в рамках аксиоматики классической теории экстремальных задач.

Некорректность задач многокритериального выбора обуславливает необходимость использования для ее решения соответствующих этапу классу задач методов. Известно, что основу таких методов составляет регуляризация-доопределение (уточнение) задачи путем привлечения дополнительной качественной и количественной информации о свойствах критериальных функций, об альтернативах, о принципах оптимальности и т.п. В рассматриваемом примере на основе дополнительной информации должен быть доопределен принцип оптимальности и методы его реализующие таким образом, чтобы в регуляризованной задаче выполнялись все условия корректности.

Вторая особенность задач многокритериального выбора состоит в том, что основным источником дополнительной информации при поиске наилучших альтернатив являются эксперты (Э), хорошо знающие заданную предметную область, и лицо, принимающее решение, преследующее определенную цель (цели), в интересах достижения которой и решается рассматриваемая задача. ЛПР, как и эксперты, должно быть компетентным специалистом в соответствующей предметной области, обладать опытом деятельности в ней, а также должно быть наделено необходимыми полномочиями.

Следует отметить, что в ряде случаев дополнительная информация в задачах многокритериального выбора может быть получена и от других источников (например, на основе анализа результатов системного моделирования)

Характерные особенности задач многокритериального выбора

Третья особенность задач многокритериального выбора заключается в том, что в данных задачах множество допустимых альтернатив и множество частных отношений предпочтений может задаваться как непосредственно, так и с использованием соответствующих функций, функционалов, операторов и т.п. Возможен комбинированный (смешанный) вариант задания множества допустимых альтернатив и отношений предпочтения. Отметим, что очень часто критериальные функции имеют различные масштабы измерения и их сравнение становится трудным, а в ряде ситуаций даже невозможным. Поэтому данные критериальные функции необходимо приводить к единому масштабу измерения или, другими словами, нормализовать их.

Таким образом, основные особенности и соответствующие проблемы, связанные с решением задач многокритериальной оптимизации, имеют скорее не вычислительный, а концептуальный характер, т.к. невозможно строго математически доказать, что выбранная в конкретных условиях ЛПР альтернатива из числа недоминируемых (неулучшаемых одновременно по всем показателям) является наилучшей. В другой ситуации ЛПР может выбрать другую недоминируемую альтернативу. Указанное положение можно считать основной аксиомой в задачах принятия многокритериальных решений.

Для корректного решения на практике перечисленных выше проблем необходимо уметь строить математические модели многокритериальной оптимизации и обоснованно применять для поиска «наилучших» альтернатив соответствующие методы и алгоритмы оптимального выбора.

Первым шагом в процессе построения указанных математических моделей является их обобщенное теоретико-множественное описание.

Уточненное описание структуры выбора с многими отношениями предпочтения. Общая постановка задач векторной оптимизации

Обобщенная структура выбора с мультипредпочтением, описывающая задачи векторной оптимизации, имеет следующий вид:

$$\langle \Delta_{s\beta}, \{r_i^\alpha\}_{i \in \Gamma}, \{F_k\}_{k \in \Gamma_1} \rangle$$

где $\Delta_{s\beta}$ — множество допустимых альтернатив;

$\{r_i^\alpha\}_{i \in \Gamma}$ — множество исходных отношений предпочтения;

$\{F_k\}_{k \in \Gamma_1}$ — множество правил согласований отношений предпочтения.

Уточненное описание структуры выбора с многими отношениями предпочтения. Общая постановка задач векторной оптимизации

В задачах векторной оптимизации исходные (входные) отношения предпочтения r_i^α задаются посредством функций (функционалов) $f_i : \Delta \rightarrow H_i, H_i \subseteq \mathbb{R}^1 (i \in \Gamma)$, отображающих H_i множество альтернатив на подмножество действительной оси и дающих каждой альтернативе кардинальную (количественную) оценку. Подмножество $H_i (i \in \Gamma)$ называется шкалой оценок по критериальной функции f_i . В дальнейшем в данной главе ограничения общности будем предполагать, что каждую критериальную функцию f_i необходимо максимизировать (т.е. большие значения критериальных функций предпочтительнее меньших), а предпочтения ЛПР не меняются скачком. Кроме того, для удобства в дальнейшем вектор будем обозначать просто символом x .

Если множество Γ состоит из « m »-элементов ($\Gamma = \{1, 2, \dots, m\}$), то функции образуют « m »-мерный вектор $f = \|f_1, f_2, \dots, f_m\|$ сопоставляющий, каждой «точке» принадлежащей пространству (области) допустимых альтернатив $x \in \Delta_{s\beta}$ соответствующую «точку» в m -мерном пространстве целевых (критериальных) функций $f(x) \in \mathbb{R}^m$.

Уточненное описание структуры выбора с многими отношениями предпочтения. Общая постановка задач векторной оптимизации

На рис.4.1 для случая $m=n=2$ приведена используемая обычно для пояснения сущности проблем многокритериального выбора геометрическая интерпретация пространства допустимых альтернатив $\Delta_{s\beta}$ и пространства целевых (критериальных) функции $\Phi, f = \|f_1, f_2\|^T \in \Phi$.

Конечной целью исследования задач векторной оптимизации обычно является отыскание некоторой наилучшей (оптимальной, эффективной) альтернативы, принадлежащей множеству допустимых альтернатив $\Delta_{s\beta}$. При этом в настоящее время известно большое разнообразие вариантов задания множества $\Delta_{s\beta}$, каждый из которых соответствует конкретной модели, относящейся, например, к классу математических, логико-лингвистических либо логико-алгебраических моделей.

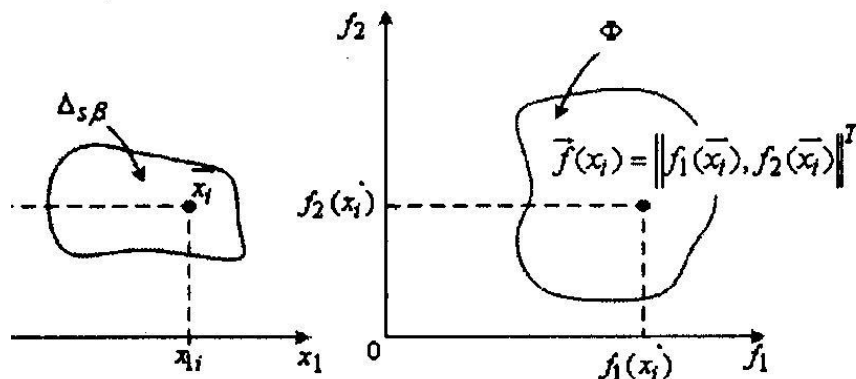


Рис. 4.1.

Частные показатели качества программных средств

1. *Липаев В.В. Надежность программных средств. М.:Синтез, 1998.*
2. *Липаев В.В. Оценка качества программных изделий. М.:Эрис, 2001.*
3. *Липаев В.В. Обеспечение качества программных средств. Методы и стандарты. М.:Синтез, 2001.*
4. *Липаев В.В. Качество программных средств. М.:Янус, 2002.*
5. *Международный стандарт ISO 9126:1991 «Информационная технология. Оценка программного продукта. Характеристики качества и руководство по их применению».*
6. *Пальчун Б.П., Юсупов Р.М. Оценка надежности программного обеспечения. СПб.:Наука, 1991.*
7. *Баранов С.Н., Домарацкий А.Н., Ласточкин Н.К., Морозов В.П. Процесс разработки программных изделий. М.:Наука, 2000.*

Список основной рекомендуемой литературы

1. Липаев В.В. Надежность программных средств. М.:Синтез, 1998.
2. Липаев В.В. Оценка качества программных изделий. М.:Эрис, 2001.
3. Липаев В.В. Обеспечение качества программных средств. Методы и стандарты. М.:Синтез, 2001.
4. Липаев В.В. Качество программных средств. М.:Янус, 2002.
5. Международный стандарт ISO 9126:1991 «Информационная технология. Оценка программного продукта. Характеристики качества и руководство по их применению».
6. Пальчун Б.П., Юсупов Р.М. Оценка надежности программного обеспечения. СПб.:Наука, 1991.
7. Баранов С.Н., Домарацкий А.Н., Ласточкин Н.К., Морозов В.П. Процесс разработки программных изделий. М.:Наука, 2000.

Список дополнительной рекомендуемой литературы

- МЭК 60050-191 Международный электротехнический словарь - Часть 191: Надежность и качество услуг, Редакция 2.0)
- ИИЕЕ 610.12-1990 Глоссарий по терминологии программной инженерии
- ИИЕЕ 1517-1999 (R2004), Стандарт ИИЕЕ по информационной технологии - Процессы жизненного цикла программного обеспечения - Процессы повторного использования
- ИСО/МЭК 2382-1:1993 Информационные технологии - Словарь - Часть 1: Основные термины
- ИСО/МЭК 2382-14:1997 Информационные технологии - Словарь - Часть 14: Надежность, сопровождаемость и готовность
- ИСО/МЭК 2382-20:1990 , Информационные технологии - Словарь - Часть 20: Разработка системы
- ИСО 7498-2:1989 Системы обработки информации - Взаимодействие открытых систем - Базовая эталонная модель - Часть 2: Архитектура безопасности
- ИСО/МЭК 15288:2008 Системная и программная инженерия - Процессы жизненного цикла систем
- ИСО/МЭК/ИИЕЕ 24765:2010 Системная и программная инженерия - Словарь
- ИСО/МЭК 25000:2005 Программная инженерия - Требования и оценка качества программной продукции (SQuaRE) - Руководство по SQuaRE
- ИСО/МЭК 25012:2008 Программная инженерия - Требования и оценка качества программной продукции (SQuaRE) - Модель качества данных
- ИСО/МЭК 25020:2007 Программная инженерия - Требования и оценка качества программной продукции (SQuaRE) - Эталонная модель и руководство по измерениям
- ИСО/МЭК 25030:2007 Программная инженерия - Требования и оценка качества программной продукции (SQuaRE) - Требования к качеству
- ИСО/МЭК 25040:2011 Системная и программная инженерия - Требования и оценка качества систем и программного обеспечения (SQuaRE) - Процесс оценки
- ИСО/МЭК ТО 25021:2007 Программная инженерия - Требования и оценка качества программной продукции (SQuaRE) - Элементы показателя качества
- Северный Техас, Консорциум ориентированных на сеть систем (2008), Определения надежности

Список дополнительной рекомендуемой литературы

ИСО 9001:2000 Системы менеджмента качества - Требования

ИСО/МЭК 9126-1:2001 Программная инженерия - Качество продукта - Часть 1: Модель качества

ИСО/МЭК ТО 9126-2:2003 Программная инженерия - Качество продукта - Часть 2: Внешние показатели

ИСО/МЭК ТО 9126-3:2003 Программная инженерия - Качество продукта - Часть 3: Внутренние показатели

ИСО/МЭК ТО 9126-4:2004 Программная инженерия - Качество продукта - Часть 4: Показатели качества при использовании

ИСО 9241-11:1998 Эргономичные требования для офисной работы с терминалами визуального представления (VDTs) - Часть 11: Руководство по удобству использования

ИСО 9241-14:1997 Эргономичные требования для офисной работы с терминалами визуального представления (VDTs) - Часть 14: Диалоги меню

ИСО 9241-110:2006 Эргономика взаимодействия человек-система - Часть 110: Принципы диалога

ИСО/МЭК 12207:2008 Системная и программная инженерия - Процессы жизненного цикла программного обеспечения

ИСО/МЭК 13335-1:2004 Информационные технологии - Методы и средства обеспечения безопасности - Менеджмент безопасности информационно-коммуникационных технологий - Часть 1: Понятия и модели менеджмента безопасности информационно-коммуникационных технологий

ИСО 13407:1999 Процессы проектирования для интерактивных систем, ориентированные на человека

ИСО/МЭК 14598-2:2000 Программная инженерия - Оценка программного продукта - Часть 2: Планирование и управление

ИСО/МЭК 14598-3:2000 Программная инженерия - Оценка программного продукта - Часть 3: Процесс для разработчиков

ИСО/МЭК 14598-4:1999 Программная инженерия - Оценка программного продукта - Часть 4: Процесс для заказчиков

ИСО/МЭК 14598-5:1998 Информационные технологии - Оценка программного продукта - Часть 5: Процесс для оценщиков

ИСО/МЭК 14598-6:2001 Программная инженерия - Оценка программного продукта - Часть 6: Документация модулей оценки

ИСО/МЭК 15026:1998 Информационные технологии - Уровни целостности систем и программного обеспечения

Acknowledgement

The research described in this paper is partially supported by International project ERASMUS +, Capacity building in higher education, № 73751-EPP-1-2016-1-DE-EPPKA2-CBHE-JP, Innovative teaching and learning strategies in open modelling and simulation environment for student-centered engineering education.

Контактная информация

Соколов Борис Владимирович:

❖ **Phone: +7 812 328-23-76;**

❖ **Fax: +7 812 328-44-50;**

❖ **E-mail: sokol@iias.spb.su;**

**Web: <http://www.spiiras-gro>Web:
<http://www.spiiras-grom.ru>**

СПАСИБО ЗА ВНИМАНИЕ