

# Jet Info

ИНФОРМАЦИОННЫЙ БЮЛЛЕТЕНЬ

№ 3 (130)/2004



**Технологические  
процессы и стандарты  
обеспечения  
функциональной  
безопасности  
в жизненном цикле  
программных средств**

ИНФОРМАЦИОННАЯ  
БЕЗОПАСНОСТЬ

*Рассмотрены содержание комплекса базовых стандартов, регламентирующих процессы обеспечения функциональной безопасности в жизненном цикле сложных программных средств и систем реального времени. Обращается внимание на близость состава процессов, применяемых для обеспечения безопасности, и целесообразность компоновать на их основе нормативные документы для реализации конкретных проектов комплексов программ.*

## Введение

Высокое качество при создании объектов и систем может быть обеспечено измерением их реальных характеристик и отбраковкой не удовлетворяющих требованиям, или применением высококачественных технологий и процессов, поддерживающих их жизненный цикл (ЖЦ). В этом случае не всегда известно, какое качество объектов достигнуто, но приложены все соответствующие, возможные и доступные методы и усилия, чтобы качество было максимально высоким. Прямое измерение высокой достигнутой функциональной безопасности объектов и систем сталкивается с большими трудностями, прежде всего вследствие необходимости фиксировать редкие, непредсказуемые отказовые ситуации с неожиданным ущербом. Поэтому основное содержание стандартов и нормативных документов в области функциональной безопасности составляют методы, процессы и технологии обеспечения высокой безопасности систем в жизненном цикле и почти не уделяется внимание определению при этом значений достигаемой безопасности. Соответственно усилия разработчиков систем и программных средств сосредотачиваются на методах регламентирования и обеспечения качества жизненного цикла таких объектов и систем.

Встроенные программные средства (ПС), в частности, реализующие функции обеспечения безопасности управления динамическими объектами

и системами, используемыми в авиации, космосе, на транспорте, создаются с применением технологий и инструментальных средств, по существу, аналогичных применяемым для разработки других классов сложных ПС реального времени. Наиболее общими и широко применяемыми стандартами, регламентирующими процессы жизненного цикла крупных ПС высокого качества, являются **ISO 12207, ISO 15504** и целая серия стандартов, детализирующих и конкретизирующих базовые процессы этих стандартов. Однако в этих стандартах непосредственно безопасности и защите уделяется мало внимания, и специфика обеспечения функциональной безопасности сложных ПС не выделяется и не комментируется. В то же время эти стандарты целесообразно адаптировать и применять при создании ПС, реализующих функции безопасности, что в частности поддерживается ссылками на них в стандартах, посвященных непосредственно безопасности ПС, объектов и систем, реализуемых на базе ЭВМ.

Базовым задачам обеспечения безопасности программных средств и систем с использованием ЭВМ посвящена значительная группа стандартов, среди которых следует выделить **ISO 15408, ISO 13335, ГОСТ Р 51904**. Рекомендуемые процессы и структура жизненного цикла ПС в этих стандартах несколько различаются, однако по составу процессов они весьма близки. В них акцентируется внимание на **информационной безопасности**, а также на обеспечении ее качества в жизненном цикле систем обработки информации и управления. Значительная часть содержания этих стандартов уделена требованиям, методам и процессам разработки и всего жизненного цикла программных средств и систем, которые, в частности, могут использоваться для обеспечения безопасности. Многие рекомендации этих стандартов могут быть адаптированы и полезны для обеспечения **функциональной безопасности**, в том числе, встроенных ПС. Целиком проблему функциональной безопасности ПС и систем отражают стандарты **IEC 61508-3, ГОСТ Р 51904** и **IEC 60880**, в которых, как и в остальных, значительная часть содержания посвящена регламентированию процессов жизненного цикла сложных комплексов программ [1, 2]. Из них для обеспечения функциональной безопасности ЖЦ встроенных комплексов программ целесообразно применять, в основном, первые два стандарта.

Значительно более полно, систематично и подробно процессы ЖЦ сложных ПС представлены в базовых стандартах **ISO 12207, ISO 15504**, и в свыше десяти сопутствующих им стандартах, детализирующих и углубляющих требования и процессы ЖЦ ПС [3, 4]. Этот комплекс стандартов целесо-

образно учитывать при создании и применении ПС и систем, обеспечивающих функциональную безопасность. Поэтому ниже приведены краткие аннотации этих двух стандартов (см. п. 4 и 5), которые дают представление об особенностях, рекомендуемых в них процессов жизненного цикла ПС и систем. Содержание большинства представленных в данной главе стандартов весьма близко по существу, однако значительно различается терминологией и структурой.

## 1. Процессы обеспечения функциональной безопасности программных средств в стандарте IEC 61508

Технология создания и всего жизненного цикла комплексов программ для **обеспечения функциональной безопасности специализированных ЭВМ**, встроенных в аппаратуру объектов и систем наиболее четко представлена в стандарте IEC 61508:1-6:1998 Функциональная безопасность электрических / электронных / программируемых электронных систем безопасности. Третья часть стандарта IEC 61508-3 "Требования к программному обеспечению" устанавливает общий подход ко всем видам деятельности на протяжении цикла обеспечения безопасности для систем, содержащих программируемые электронные компоненты (ПЭС), которые используются для выполнения различных функций и, в частности, для обеспечения безопасности систем. Этот унифицированный подход рекомендован с целью выработки последовательной политики безопасности для всех систем, основанных на ЭВМ. Любая стратегия функциональной безопасности должна учитывать не только все элементы в каждой конкретной системе (например, датчики, управляющие устройства и исполнительные механизмы), но, также, все системы и программные

средства, обеспечивающие безопасность, и создавать суммарную комбинацию систем, обеспечивающих безопасность. Стандарт IEC 61508-3 содержит следующие особенности [5]:

- рассматривает все этапы циклов обеспечения общей функциональной безопасности ПЭС и программных средств;
- учитывает быстро развивающиеся технологии; его структура достаточно прочная и всеобъемлющая для того, чтобы обслуживать будущие разработки;
- позволяет разработать область применения международных стандартов; обеспечивает высокий уровень согласованности (например, основополагающих принципов, терминологии и т.п.) как внутри каждой области применения ПЭС, так и между областями применения;
- обеспечивает метод разработки спецификаций требований по безопасности ПЭС, необходимых для достижения заданной функциональной безопасности;
- использует основанный на рисках подход для определения требований к уровням соответствия комплексу требований по функциональной безопасности;
- устанавливает количественные меры отказов для систем безопасности ПЭС, связанные с уровнями соответствия комплексу требований по функциональной безопасности;
- программные средства, обеспечивающие безопасность, включают операционные системы, системное программное обеспечение, программное обеспечение в коммуникационных сетях, функции интерфейса человек-машина, инструментальные средства поддержки и программно-аппаратные средства, а также прикладные программы;
- устанавливает требования для этапов жизненного цикла обеспечения безопасности и деятельности, которая должна проводиться при проектировании и разработке ПС, обеспечивающего безопасность (модель жизненного цикла обеспечения безопасности);
- обеспечивает требования для информации, относящейся к аттестации ПС, которая должна передаваться организациям, производящим компоновку ПЭС в составе системы;
- обеспечивает требования по подготовке информации и процедур, относящихся к ПС, необходимых пользователю для эксплуатации и обслуживания систем безопасности;
- обеспечивает требования, которые должны выполнять организации, производящие модификацию ПС, обеспечивающего безопасность;

- обеспечивает требования к технологическим средствам поддержки, таким как инструментальные средства проектирования и разработки, языковые трансляторы, инструментальные средства для тестирования, отыскания и устранения ошибок, средства управления конфигурацией.

Планирование функциональной безопасности ПС должно определять стратегию получения, разработки, компоновки, верификации, аттестации и модификации комплекса программ в той мере, которая необходима для требуемого уровня соответствия ПЭС комплексу требований по безопасности.

## Требования к циклу обеспечения безопасности программных средств

### 1. Общие положения.

Цикл обеспечения безопасности при разработке ПС должен быть выбран и задан при планировании безопасности, приспособлен для практических нужд проекта или организации. В деятельность, связанную с циклом обеспечения безопасности, должны быть включены процедуры гарантирования качества и безопасности. Каждый этап цикла обеспечения безопасности ПС должен быть подразделён на элементарные операции в рамках сферы действия, а также входных и выходных данных, оговоренных для каждого этапа. Полный перечень этапов цикла обеспечения безопасности подходит для больших вновь разрабатываемых систем. В малых системах может оказаться целесообразным объединять этапы проектирования ПС и архитектурного проектирования объекта или системы. Рекомендуется использовать основные положения, комплекс этапов и работ, представленный в стандарте **ISO 12207**. Для каждого этапа цикла обеспечения безопасности должны использоваться соответствующие методы и меры. По результатам деятельности в рамках цикла обеспечения безопасности ПС должна быть составлена документация. Если на любом этапе обеспечения безопасности ПС потребуется изменение, относящееся к более раннему этапу, то более ранний и последующие этапы цикла обеспечения безопасности должны быть повторены.

### 2. Спецификация требований по безопасности программного средства.

Спецификация должна основываться на заданных требованиях по безопасности системы, общего и специального прикладного ПС. Спецификация требований по безопасности должна быть доста-

## Требования к циклу обеспечения безопасности программного средства

1. Общие положения
2. Спецификация требований по безопасности программного средства
3. Планирование аттестации программного средства
4. Проектирование и разработка программного средства:
  - 4.1. Цели
  - 4.2. Общие требования
  - 4.3. Требования к архитектуре программного средства
  - 4.4. Требования к инструментальным средствам поддержки и языкам программирования
  - 4.5. Требования к рабочему проекту и разработке
  - 4.6. Требования к кодированию
  - 4.7. Требования к испытаниям модулей программного средства
  - 4.8. Требования к компоновочным испытаниям программного средства
5. Компоновка программируемой электроники
6. Процедуры эксплуатации и обслуживания программного средства обеспечения
7. Аттестация программного средства
8. Модификация программного средства
9. Верификация программного средства

## Оценка функциональной безопасности

Рис. 1.

точно подробной для того, чтобы конструкция и ее выполнение достигали требуемого соответствия комплексу требований по безопасности, и можно было произвести оценку функциональной безопасности. В частности, разработчик ПС должен учитывать: функции безопасности; конфигурацию или архитектуру системы; производительность и время срабатывания; взаимодействие между оборудованием и оператором. Эти требования в большинстве случаев должны достигаться комбинацией основного функционального программного средства и специального ПС обеспечения безопасности. Точное разделение между основным и специальным прикладным ПС зависит от выбранной архитектуры комплекса программ. В спецификации должны быть **требования к функциям безопасности**:

- обеспечивающим достижение и поддержание безопасного состояния объекта или системы;

- относящимся к выявлению отказов; извещению о их наличии и управлению отказами в аппаратуре программируемой электроники; отказами датчиков и исполнительных механизмов, а также в самом программном средстве;
- относящимся к периодической проверке функций обеспечения безопасности в оперативном режиме и в отключённом состоянии;
- производительности и времени срабатывания основных функций ПС.

### 3. Планирование аттестации программного средства.

Должно быть произведено планирование с целью определения шагов, как процедурных, так и технических, которые должны быть использованы для демонстрации того, что программное средство удовлетворяет требованиям по его безопасности. План аттестации безопасности ПС должен содержать следующее: подробные указания о том, когда должна производиться аттестация; кто должен производить аттестацию; определение ответственных режимов эксплуатации технических средств; техническую стратегию аттестации; требуемые условия окружающей среды, в которой должна производиться аттестация; критерии прохождения / не прохождения; политику и процедуры оценки результатов аттестации.

### 4. Проектирование и разработка программного обеспечения.

**4.1. Цели.** Создание архитектуры программного средства, которая отвечает заданным требованиям по безопасности ПС, рассмотрение и оценка требований, возлагаемых на ПС со стороны архитектуры аппаратуры системы безопасности, включая взаимодействие аппаратуры и программного средства для обеспечения безопасности объекта или системы. Выбор подходящего для требуемого уровня соответствия комплексу требований по безопасности, комплекта инструментальных средств (включая языки и компиляторы), который способствовал бы проведению верификации, аттестации, оценки и модификации на протяжении всего цикла обеспечения безопасности ПС. Проектирование и изготовление ПС, отвечающего заданным требованиям по безопасности в отношении требуемого уровня соответствия комплексу требований по безопасности системы, которое можно проанализировать и проверить и безопасно модифицировать.

**4.2. Общие требования.** В соответствии с требуемым уровнем комплекса требований по безо-

пасности, выбранный метод проектирования должен обладать свойствами, которые способствуют: абстрагированию, модулированию и другим методам, влияющим на сложность разработки. Выбранный метод проектирования должен обладать свойствами, облегчающими модификацию программного обеспечения. Такие свойства включают модульность, сокрытие информации и формирование пакетов данных. Насколько это возможно практически, конструкция должна сводить к минимуму ту часть ПС, которая относится к обеспечению безопасности. Конструкция должна включать функции программного средства по проведению контрольных испытаний и всех диагностических процедур с тем, чтобы выполнять требования по соответствию комплексу требований по безопасности системы. Конструкция ПС должна включать соответствующий уровню комплекса требований по безопасности, самоконтроль потока управления и потока данных. При выявлении неисправностей должны предприниматься соответствующие контрмеры.

**4.3. Требования к архитектуре программного средства.** Архитектура ПС определяет основные компоненты и подсистемы комплекса программ, как они соединяются между собой, и как могут быть достигнуты требуемые свойства, в частности, соответствие комплексу требований по безопасности. Основные компоненты ПС включают: операционные системы, базы данных, внутренние подсистемы ввода/вывода, коммуникационные подсистемы, прикладные программы, инструментальные средства программирования и диагностики и т.п. В крайнем случае, с использованием языков широкого применения, архитектура системы должна создаваться поставщиком специально для этого применения (или этого класса применений). С точки зрения безопасности создание архитектуры ПС является этапом, на котором разрабатывается стратегия основной безопасности комплекса программ и системы. Конструкция архитектуры программного средства должна устанавливаться поставщиком и/или разработчиком, при этом должно быть создано подробное описание конструкции. Описание должно быть основано на подразделении на компоненты/подсистемы, для каждой из которых предоставлена следующая информация:

- являются ли они новыми, существующими или собственными;

- проходили ли они ранее верификацию, и если да, условия проведения их верификации;
- относится ли каждая подсистема/компонент к обеспечению безопасности или нет;
- уровень соответствия ПС комплексу требований по безопасности для каждой подсистемы/компонента;
- определять все взаимодействия ПС и аппаратуры и подробно оценивать их значение;
- использовать для представления архитектуры систему обозначений, которая однозначно определяет или ограничивается однозначно определяемыми свойствами.

**4.4. Требования к инструментальным средствам поддержки и языкам программирования.** Выбор инструментальных средств разработки зависит от характера деятельности по разработке и от архитектуры ПС. Для требуемого уровня безопасности должен быть выбран соответствующий комплект технологических средств, включая языки, компиляторы, инструментальные средства управления конфигурацией, и при необходимости, автоматические испытательные средства. Нужно рассмотреть наличие подходящих инструментальных средств (не обязательно таких, какие применялись при первоначальной разработке системы) для обеспечения необходимых действий на протяжении всего срока службы системы безопасности.

**4.5. Требования к рабочему проекту и разработке ПС.** Характер рабочего проекта и процесса разработки может изменяться в зависимости от архитектуры ПС. В случае прикладного программирования пользователем с использованием языка ограниченного применения, например, многоступенчатой логики и функциональных блоков, рабочий проект может рассматриваться скорее как конфигурирование, а не программирование. Однако считается хорошей практикой проектировать программы структурным методом, в том числе организация ПС в виде модульной структуры, которая подразделяется (насколько это возможно) на части, относящиеся к обеспечению безопасности, которые обеспечивают защиту от дефектов и ошибок при введении новых данных; использовании уже апробированных модулей, и выполнении конструкций, которые облегчают будущие модификации ПС. В описании архитектуры ПС дальнейшие уточнения конструкции каждого компонента/подсистемы должны основываться на подразделении программ на модули. Должны быть оговорены конструкция каждого модуля и испытания, относящиеся к

модулю ПС. Должны быть установлены соответствующие компоновочные испытания системы для доказательства того, что комплекс программ отвечает заданным требованиям по безопасности.

**4.6. Требования к кодированию.** Исходный код должен быть читаемым, понятным и поддающимся проверке; удовлетворять заданным требованиям для конструкции модулей ПС; удовлетворять заданным требованиям стандартов кодирования; удовлетворять всем требованиям, заданным при планировании безопасности.

**4.7. Требования к испытаниям модулей программного средства.** Каждый модуль должен быть испытан, как это оговорено при проектировании конструкции ПС. Эти испытания должны показать, что каждый модуль выполняет предназначенную ему функцию и не выполняет не предназначенных функций. По результатам испытаний модуля ПС должна быть составлена документация. Должна быть оговорена процедура корректировки в случае неудачного прохождения испытаний.

**4.8. Требования к компоновочным испытаниям программного средства.** Комплекс программ должен быть протестирован в соответствии со спецификацией квалификационных испытаний. Эти испытания должны подтвердить, что все модули и компоненты/подсистемы ПС взаимодействуют правильно и выполняют предназначенные им функции и не выполняют не предназначенных функций. По результатам компоновочных испытаний должна быть составлена документация, содержащая результаты испытаний, а также заключение, достигнуты ли все цели и выполнены ли все критерии испытаний. Если испытания прошли неудачно, в документации должны быть указаны причины. В процессе компоновки ПС любая модификация или изменение должны быть проанализированы с точки зрения их влияния, при анализе должны быть выявлены все подвергшиеся воздействию модули и произведены необходимые действия по их корректировке, повторной верификации и изменению конструкции.

## 5. Компоновка программируемой электроники.

Целью требований является встраивание комплекса программ в заданную аппаратуру, соединение ПС и аппаратуры в программируемую электронику, обеспечивающую безопасность, их совместимость и удовлетворение требования заданного

уровня по безопасности. При компоновочных испытаниях аппаратуры и программного средства любая модификация или изменение объединённой системы должны анализироваться с точки зрения взаимодействия, при котором должны быть выявлены подвергающиеся воздействию модули ПС и проведена необходимая деятельность по повторной верификации. По компоновочным испытаниям должна быть составлена документация, содержащая результаты испытаний и заключение о том, достигнуты ли цели и критерии испытаний. Если испытания прошли неудачно, в документации должны быть указаны причины этого.

## 6. Процедуры эксплуатации и обслуживания программного средства.

Установление информации и относящихся к ПС процедур, необходимых для подтверждения того, что функциональная безопасность системы сохраняется в процессе эксплуатации и модификации. В данном стандарте программное средство (в отличие от аппаратуры) невозможно оперативно обслуживать, оно всегда только модифицируется.

## 7. Аттестация программного средства.

Целью требований является обеспечение соответствия объединённой системы, заданным требованиям по безопасности ПС при заданном уровне безопасности внешней среды и системы. Аттестация должна производиться, как это оговорено при планировании аттестации программного обеспечения. Для каждой функции безопасности в документации по аттестации ПС должны быть указаны следующие результаты: использованный вариант плана аттестации; функция, прошедшая аттестацию (путём анализа, экспертизы или экспериментальных испытаний) вместе со ссылками на план аттестации ПС; инструментальные средства и оборудование, и данные об их калибровке; результаты аттестации; расхождение между ожидаемыми и полученными результатами. Если обнаружены расхождения между ожидаемыми и полученными результатами, следует принять решение о том, продолжать ли аттестацию или сделать заявку на изменения и вернуться к более ранней части разработки цикла обеспечения безопасности, для этого составляется документация, являющаяся частью результатов аттестации программного обеспечения. Основным методом аттестации ПС должны быть экспериментальные испытания; синтез динамических изображений и моделирование могут использоваться дополнительно; программное средство должно быть проверено тестированием при имитации:

- входных сигналов, существующих при нормальной эксплуатации;

- ожидаемых происшествий и аномалий;
- нежелательных ситуаций, требующих вмешательства системы безопасности.

Всё оборудование, используемое для аттестации, должно быть квалифицировано в соответствии со спецификацией, согласующейся со стандартом или общепринятой процедурой. К результатам аттестации ПС предъявляются следующие требования: испытания должны показать, что все заданные требования по безопасности программного средства выполняются правильно, и что ПС не выполняет не предназначенных ему функций. По каждому испытанию и его результатам должна быть составлена документация для проведения последующего анализа и независимой оценки соответствия с требуемым уровнем безопасности.

## 8. Модификация программного средства.

Целью является производство корректировок, улучшение или приспособление аттестуемого ПС для гарантии поддержания требуемого уровня соответствия комплексу требований по безопасности. Модификация может производиться только при наличии санкционированной заявки на изменения в соответствии с процедурами, установленными при планировании совершенствования безопасности. Все модификации, оказывающие влияние на функциональную безопасность системы, должны вызывать возврат к соответствующему этапу обеспечения безопасности ПС. Все последующие этапы должны быть повторены в соответствии с процедурами, установленными для конкретных этапов в соответствии с требованиями стандарта. Планирование безопасности при модификации ПС, относящегося к обеспечению безопасности, должно включать следующую информацию: определение персонала и спецификацию его требуемой компетентности; детальную спецификацию модификации; планирование верификации; объём повторной аттестации и испытаний модификации в пределах, требуемых уровнем соответствия комплексу требований по безопасности. Все подробности модификации должны быть изложены в документации. Они могут использоваться на этапах компоновки программируемой электроники, общей установки и ввода в эксплуатацию.

## 9. Верификация программного средства.

Целью является испытание и оценка выходных данных этапа ЖЦ обеспечения безопасности ПС для подтверждения правильности и согласованности с выходными данными и стандартами, которые использовались в качестве входных данных на этом этапе. Основные аспекты верификации являются

общими для нескольких этапов цикла обеспечения безопасности. В этом пункте не содержатся дополнительные требования к проводимым при верификации испытаниям, изложенным в 4.7 (испытания модуля программного обеспечения), 4.8 (компоновка программного обеспечения) и 5 (компоновка программируемой электроники) (см. рис. 1), которые сами по себе являются верификационными действиями. Этот пункт не требует также проведение верификации в дополнение к аттестации программного средства (см. п. 7), которая в данном стандарте является демонстрацией соответствия системы спецификации требований по безопасности (сквозная верификация).

Верификация ПС должна планироваться одновременно с разработкой для каждого этапа обеспечения безопасности и эта информация должна быть внесена в документацию. При планировании верификации ПС должны быть указаны критерии, методы и инструментальные средства, которые должны использоваться во время верификации. При верификации должны выполняться следующие действия:

- верификация требований по безопасности программного средства;
- верификация архитектуры программного средства, конструкции системы и встроенного программного средства;
- верификация конструкции модулей ПС, объектного кода и данных;
- испытания модулей программного средства;
- компоновочные испытания программного средства;
- компоновочные испытания комплекса программируемой электроники;
- тестирование реализации требований по безопасности программного средства и системы.

## Оценка функциональной безопасности программного средства

Выбор методов оценки не гарантирует сам по себе, что будет достигнуто соответствие комплексу требований по безопасности. Производящие оценку специалисты должны учитывать:

- обоснованность и согласованность выбранных методов, языков и инструментальных средств со всем циклом разработки;
- используют ли разработчики методы, которые они полностью понимают;
- хорошо ли методы, языки и инструментальные средства подходят для исключения дефектов и ошибок, возникающих при разработке.

Для каждого метода в таблицах стандарта приводятся рекомендации по уровням соответствия комплексу требований по безопасности. Подходящие методы должны быть выбраны в соответствии с уровнем соответствия комплексу требований по безопасности. Оценка методов связана с понятием **эффективность**. При всех прочих равных факторах методы, обозначенные HR, являются более эффективными с точки зрения предотвращения систематических отказовых ситуаций при разработке программного средства или (в случае архитектуры) более эффективными с точки зрения контроля за остаточными дефектами в ПС, выявляемыми в процессе исполнения. Для конкретных областей применения соответствующая комбинация методов или мер должна быть указана при планировании безопасности.

**Функциональные шаги применения ИЕС 61508-3** начинаются с получения требований к системе безопасности и соответствующей части планов безопасности от заказчика, где должны быть:

1. Оговорены требуемые функции безопасности и относящиеся к ним уровни соответствия комплексу требований по безопасности объекта или системы:
  - адресованы функции безопасности, предназначены для них компонентам системы безопасности;
  - адресованы функции программным средствам в каждом компоненте системы безопасности.
2. Определена архитектура программного средства для всех функций безопасности, адресованных ПС.
3. Совместно с поставщиком/разработчиком рассмотрена архитектура и безопасное переменное рассмотрение функций программного средства и аппаратуры.
4. Начато планирование верификации и аттестации ПС.
5. Спроектировано, разработано, проверено и протестировано ПС в соответствии с:
  - планированием безопасности;
  - уровнем соответствия ПС комплексу требований по безопасности объекта или системы;
  - циклом обеспечения безопасности ПС.
6. Завершена деятельность по верификации ПС, скомпоновано проверенное программное средство в заданную аппаратуру, параллельно разработаны относящиеся к ПС описания процедур для пользователей и обслуживающего персонала, которые они должны выполнять при эксплуатации системы.



7. Совместно с разработчиком аппаратуры аттестовано программное средство в скомпонованных объектах или системах безопасности.
8. Переданы результаты аттестации ПС системным инженерам для дальнейшей компоновки в общую систему.
9. Если во время эксплуатации потребуются модификация ПС, повторены некоторые этапы.

Для выполнения этих шагов должны выбираться методы и меры по безопасности ПС, соответствующие требуемому уровню соответствия комплексу требований по безопасности системы. Для облегчения этого выбора в стандарте разработаны таблицы, квалифицирующие различные методы и меры в зависимости от четырёх уровней соответствия комплексу требований по безопасности. Перекрёстные ссылки в таблицах являются рассмотрением каждого метода или меры в связи со ссылками на дальнейшие источники информации. Примеры применения таблиц соответствия комплексу требований по безопасности даны в приложении, а в стандарт **ИЕС 61508-7** включён вероятный подход к определению соответствия комплексу требований по безопасности для разработанного программного средства.

## 2. Особенности процессов обеспечения функциональной безопасности программных средств в стандарте ISO 15408

Наиболее мощным современным стандартом, отражающим требования и рекомендации по обеспечению безопасности систем, содержащих программные средства, является **ISO 15408 :1999 – 1-3** Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий, состоящий из трех частей. Первая часть определяет концепцию всего стандарта, а вторая

самая большая часть формализует методы и требования к **информационной безопасности**. Его третья часть полностью посвящена процессам обеспечения **доверия – (качества) компонентов** информационных систем (ИС), реализующих функции их безопасности. По существу рассматривается регламентирование технологии и процессов обеспечения жизненного цикла программных средств, создаваемых для обеспечения безопасности функционирования и применения систем. При этом акцент документа сосредоточен на **информационной безопасности** сложных программных средств ИС. Однако основные положения этой части стандарта практически полностью применимы к технологии и процессам создания ПС и обеспечению **функциональной безопасности**, в том числе для встроенных систем. Поэтому ниже в данном разделе многие положения этой части стандарта трактуются с позиции обеспечения функциональной безопасности, а **термин – доверие** применяется как понятие **качество или уверенность выполнения требования безопасности**.

**Основная концепция ISO 15408-3** – обеспечение доверия, основанное на оценке качества (активном исследовании реализации функций) продукта или системы [1]. Нарушения безопасности ПС возникают вследствие преднамеренного использования или случайной активизации уязвимостей при применении систем и ПС по назначению. Рекомендуется ряд шагов для предотвращения уязвимостей, возникающих в продуктах и системах. Уязвимости могут возникать **вследствие недостатков**:

- **требований** продукт или система могут обладать требуемыми от них функциями и свойствами, но все же содержать уязвимости, которые делают их непригодными или неэффективными в части безопасности применения;
- **проектирования** продукт или система не отвечают спецификации, и/или уязвимости, являются следствием некачественных стандартов проектирования или неправильных проектных решений;
- **эксплуатации** продукт или система разработаны в полном соответствии с корректными спецификациями, но уязвимости возникают как результат неадекватного управления при эксплуатации.

Оценка и утверждение **целей функциональной безопасности** требуется для демонстрации заказчику или пользователю, что установленные цели проекта адекватны проблеме его безопасности. Существуют цели и функции безопасности для объекта или ПС и цели безопасности для среды. Рекомендуется сопоставлять эти цели безопасности с

идентифицированными угрозами, которым они противостоят, и/или с политикой и предположениями, которым они должны соответствовать. Не все цели безопасности могут быть реализованы соответствующим объектом, так как некоторые могут зависеть от требований безопасности системы, выполняемых ее средой. В этом случае требования безопасности, относящиеся к внешней среде, необходимо ясно изложить и оценить в контексте требований к системе.

Использование требований стандарта означает, что требования могут быть четко идентифицированы, что они автономны, и применение каждого требования возможно и даст значимый результат при оценке качества, основанный на анализе соответствия объекта этому конкретному требованию. Отношение между функциями безопасности системы и функциональными требованиями может быть отношением типа "многие ко многим". Тем не менее, каждая функция безопасности должна способствовать удовлетворению, по меньшей мере, одного требования безопасности. Функции безопасности, которые не соответствуют этому положению, обычно необязательны.

**Доверие** — основа для уверенности в том, что продукт или система отвечают целям и требованиям безопасности. Активное исследование доверия — это оценка процесса функционирования системы для определения его свойств безопасности. Методы оценки могут, в частности, включать в себя:

- анализ и проверку выполнения процессов и процедур;
- проверку того, что процессы и процедуры действительно применяются;
- анализ соответствия реализации каждого положения проекта требованиям;
- верификацию доказательств правильности реализации функций;
- анализ руководств применения;
- анализ разработанных функциональных тестов и полученных результатов;
- независимое функциональное тестирование ПС и системы;
- анализ уязвимостей, включающий предположения о дефектах и ошибках.

В стандарте применяется иерархия детализации требований к безопасности и качеству систем и ПС с использованием специфических терминов: **класс** — наиболее общая характеристика качества, которая структурируется **семейством** — субхарактеристиками. Каждое семейство может иметь несколько **компонентов** — атрибутов, состоящих из **элементов** качества. Семейство доверия (качества) в стандарте может содержать один или несколько

компонентов доверия. Этот подраздел семейства доверия включает описание имеющихся компонентов и объяснение их содержания и разграничения. Подраздел идентификации компонента содержит описательную информацию, необходимую для категорирования, регистрации и ссылок на компонент. Каждый элемент представляет собой требование для выполнения. Формулировки этих требований к ПС должны быть четкими, краткими и однозначными. Поэтому каждое требование рекомендуется излагать как отдельный элемент. Структура процессов жизненного цикла систем и программных средств, обеспечивающих информационную и функциональную безопасность применения в соответствии с классами и семействами стандарта, представлена на рис. 2.

**Класс управление конфигурацией (УК)** обеспечивает сохранение целостности объектов, устанавливая и контролируя определенный порядок процессов корректировки, модификации и предоставления связанной с ними информации. УК предотвращает несанкционированную модификацию, добавление или уничтожение составляющих объектов, обеспечивая тем самым качество и документацию компонентов, которые подготовлены к распространению. Управления конфигурацией устанавливает уровень автоматизации, используемый для изменения элементов конфигурации. Возможности управления определяют характеристики системы УК. Область управления указывает на те элементы объектов, для которых необходим контроль со стороны системы управления конфигурацией.

**Класс поставка и эксплуатация** определяет требования к мерам, процедурам и стандартам, применяемым для безопасной поставки, установки и эксплуатации ПС, обеспечивая, чтобы безопасность объектов не нарушалась во время его распространения, внедрения и эксплуатации. Поставка распространяется на процедуры, используемые для поддержки безопасности во время передачи объекта пользователю при первоначальной поставке и последующих модификациях. Она включает в себя специальные процедуры, необходимые для демонстрации подлинности поставленного объекта. Такие процедуры и меры — основа обеспечения безопасности во время и после передачи ПС для применения. Установка, генерация и запуск предусматривает, чтобы копия объекта была конфигурирована и активирована администратором так, чтобы показать те же самые свойства безопасности, что и у оригинала. Эти процедуры обеспечивают уверенность в том, что администратор будет осведомлен о параметрах конфигурации объекта и о том, как они способны повлиять на функциональную безопасность.

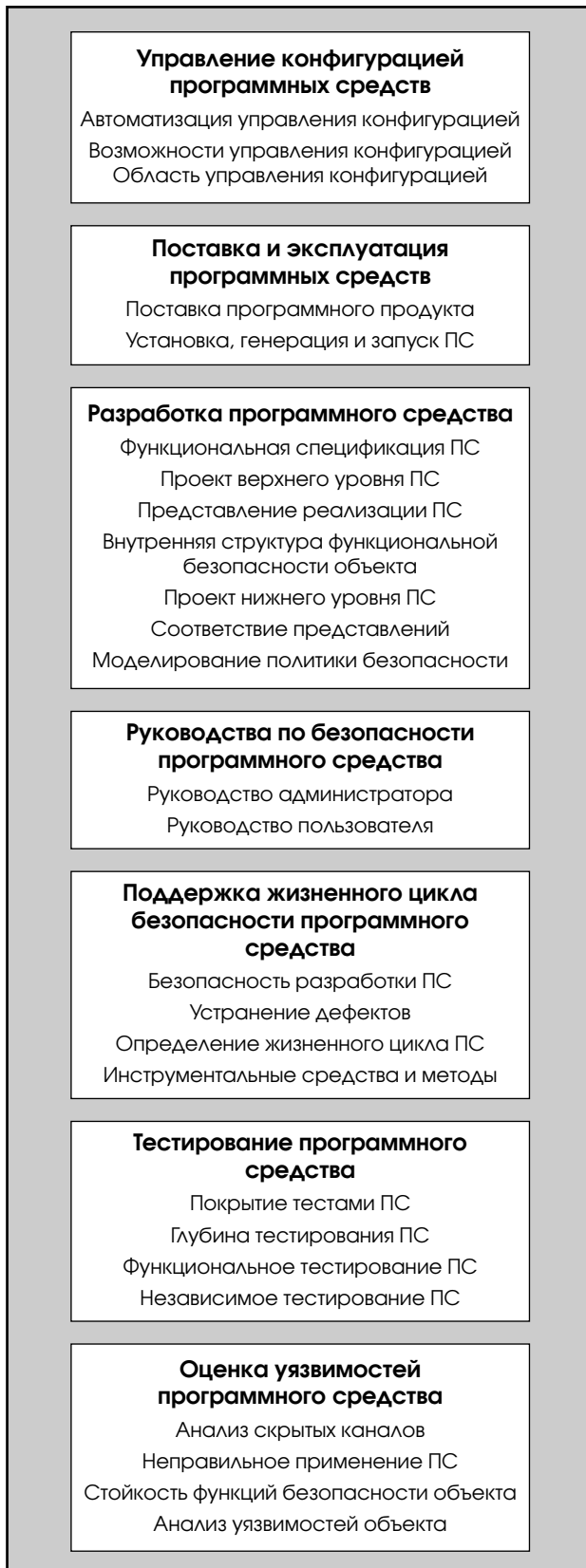


Рис. 2.

Класс **разработка** определяет требования для пошагового уточнения функциональной безопасности, начиная с краткой спецификации объекта в задании на безопасность (ЗБ) и вплоть до фактической реализации (см. рис. 2). Каждое из получаемых представлений содержит информацию, помогающую оценщику решить, были ли выполнены функциональные требования к безопасности. Функциональная спецификация описывает функции безопасности, и необходимо, чтобы она была полным и точным отображением требований безопасности объекта. Функциональная спецификация также детализирует его внешний интерфейс. Предполагается, что пользователи и заказчики объекта взаимодействуют с функциональной безопасностью через этот интерфейс.

Проект верхнего уровня – проектная спецификация самого высокого уровня, которая уточняет функциональную спецификацию безопасности системы в основных составляющих частях. Она идентифицирует базовую структуру функциональной безопасности, а также основные элементы аппаратных, программных и программно-аппаратных средств. Представление реализации – наименее абстрактное отражение функциональной безопасности. Оно фиксирует ее детализированное внутреннее содержание на уровне исходного текста, аппаратных схем и т.д. Требования к внутренней структуре определяют необходимое структурирование функций безопасности.

Проект нижнего уровня – детализированная проектная спецификация, уточняющая проект верхнего уровня до необходимой детализации, которая может быть использована как основа для программирования и/или проектирования аппаратуры и программных компонентов. Соответствие представлений – демонстрация отображения между всеми смежными парами имеющихся представлений функций безопасности, от краткой спецификации объекта до наименее абстрактного из имеющихся представлений.

Модели политики безопасности – структурные представления методов, используемые для обеспечения повышенного доверия, что функциональная спецификация соответствует принятой политике безопасности и, в конечном счете, функциональным требованиям безопасности системы. Это достигается посредством определения соответствия между функциональной спецификацией, моделью политики безопасности и моделируемыми методами обеспечения безопасности ПС.

Класс **руководства по безопасности** определяет требования, направленные на обеспечение понятности, достаточности и законченности эксплуатационной документации, представляемой разра-

ботчиком. Эта документация, которая содержит две категории информации (для пользователей и администраторов), является важным фактором безопасной эксплуатации объекта и ПС.

Требования к руководству администратора должны обеспечивать отражения ограничений среды, которые будут поняты администраторами и операторами. Руководство администратора — основной документ, имеющийся в распоряжении разработчика, для предоставления администраторам объекта, детальной и точной информации о том, как осуществлять администрирование безопасным способом и эффективно использовать доступные процедуры обеспечения безопасности.

Требования к руководству пользователя должны обеспечивать возможность эксплуатировать объект безопасным способом. Руководство — основной документ, имеющийся в распоряжении разработчика, для предоставления пользователям необходимой общей и специфической информации о том, как правильно использовать функции безопасности. В руководстве необходимо осветить два аспекта. Во-первых, требуется объяснить, что делают доступные пользователю процедуры безопасности, и как они будут использоваться, чтобы пользователи имели возможность последовательно и действенно защищать свою систему. Во-вторых, требуется разъяснить роль пользователя в поддержании безопасности системы и ПС.

Класс **поддержка жизненного цикла** определяет требования для реализации всех этапов разработки четко определенной модели, включая политики и процедуры устранения недостатков и дефектов, правильное использование инструментальных средств и методов, а также меры безопасности для защиты среды разработки.

Безопасность разработки охватывает физические, процедурные, относящиеся к персоналу и другие меры безопасности, используемые применительно к среде разработки. Она также содержит требования к физической безопасности местоположения разработки и к контролю за отбором и наймом персонала разработчиков. Устранение дефектов обеспечивает, чтобы недостатки, обнаруженные потребителями, отслеживались и исправлялись, пока объект сопровождается разработчиком. Несмотря на то, что при оценке объекта не может быть принято решение о потенциальном соответствии требованиям устранения недостатков, можно оценить политики и процедуры, которые разработчик предусмотрел для выявления и устранения дефектов и распространения исправлений потребителям.

Определение жизненного цикла ПС устанавливает, что технология разработки, используемая

разработчиком для создания объекта, включает в себя положения и действия, указанные в требованиях к процессу разработки и поддержке эксплуатации. Уверенность в соответствии объекта требованиям больше, когда анализ безопасности и подготовка свидетельств осуществляются на регулярной основе, как неотъемлемая часть процесса разработки и поддержки эксплуатации всей системы и ПС. Инструментальные средства и методы связаны с необходимостью определения средств разработки, используемых для анализа и создания объекта и ПС. Сюда включены требования, относящиеся к инструментальным средствам разработки и опциям этих инструментальных средств, зависящим от их реализации.

Класс **тестирование** устанавливает требования, которые должны демонстрировать, что реализованные функции удовлетворяют функциональным требованиям безопасности системы. Покрытие определяет полноту функциональных тестов, выполненных разработчиком для анализа качества системы и ПС. Оно связано со степенью тестирования функций безопасности. Глубина тестирования характеризуется уровнем детализации, на котором разработчик проверяет программы. Тестирование функций безопасности основано на последовательно увеличивающейся глубине информации, получаемой из анализа представлений безопасности.

Функциональное тестирование ПС должно устанавливать, что функции безопасности действительно демонстрируют свойства, необходимые для удовлетворения требований спецификации. Функциональное тестирование обеспечивает доверие, что функции удовлетворяют, по меньшей мере, требованиям выбранных функциональных компонентов. Эти процедуры сосредоточены на функциональном тестировании, выполняемом разработчиком. Независимое тестирование определяет степень выполнения функционального тестирования третьей стороной, кроме разработчика и заказчика. Эти процедуры повышают ценность тестирования добавлением тестов, которые расширяют тесты разработчика.

Класс **оценка уязвимости ПС** определяет требования, направленные на идентификацию уязвимостей, которые могут проявиться и быть активизированы. Особое внимание должно быть уделено уязвимостям, которые вносятся при проектировании, эксплуатации, неправильном применении или неверной конфигурации объекта. Анализ скрытых каналов направлен на выявление и анализ непредусмотренных коммуникационных каналов, которые могут применяться при нарушениях предписанных функций безопасности. Анализ неправильного применения позволяет выяснить, спосо-

бен ли администратор или пользователь, используя руководства, определить, что система или ПС конфигурированы или эксплуатируются небезопасным способом.

Анализ стойкости направлен на определение функций безопасности, которые реализованы с помощью вероятностного или перестановочного механизма. Даже если такие функции нельзя обойти, отключить или исказить, не исключено, что их все же можно преодолеть прямой атакой. Может быть заявлен уровень или специальная метрика стойкости для каждой из этих функций. Анализ стойкости функций выполняют для принятия решения, отвечают ли такие функции сделанным заявлениям. Анализ уязвимостей заключается в идентификации недостатков, которые могли быть внесены на различных этапах разработки. Эти потенциальные уязвимости оцениваются посредством тестирования проникновения, позволяющим сделать заключение, могут ли они в действительности быть использованы для нарушения безопасности системы и ПС.

Для систематического применения приведенных на рис. 2 классов и семейств требований в стандарте введено понятие **профиль защиты (ПЗ)** — независимая от реализации совокупность требований безопасности для некоторой категории объектов, отвечающая специфическим требованиям проекта и потребителя. Цель разработки и оценки ПЗ — показать, что он является полным, непротиворечивым, технически правильным и поэтому пригоден для изложения конкретных требований к одному или нескольким типам объектов. Оцененный ПЗ пригоден в качестве основы для разработки задания по безопасности. Для принятия решения о достаточности требований безопасности в составе ПЗ важно, чтобы решаемая задача безопасности ясно понималась всеми участниками оценки.

**Задание по безопасности (ЗБ)** — совокупность требований и спецификаций, предназначенная для использования в качестве основы для оценки конкретного объекта. Цель оценки ЗБ — показать, что оно является полным, непротиворечивым, технически правильным и поэтому пригодно для использования в качестве основы при оценке уровня безопасности соответствующей системы. Если, после тщательного рассмотрения, окажется, что ни один из компонентов требований настоящего стандарта не применим непосредственно ко всем или к части требований безопасности системы, разработчик ЗБ может сформулировать другие требования, которые не ссылаются на этот стандарт. Использование таких требований должно быть строго обосновано.

Не все перечисленные выше классы и семейства процессов обеспечения функциональной бе-

зопасности систем и ПС целесообразно применять в каждом проекте. В зависимости от сложности и критичности требования к безопасности функционирования системы и доступных ресурсов для ее реализации, стандартом рекомендуется выбирать набор классов и семейств процессов, достаточных для обеспечения необходимого качества комплекса функциональной безопасности проекта — **оценочный уровень доверия**. Оценочные уровни доверия (ОУД) образуют возрастающую шкалу достигаемого качества безопасности, которая позволяет соотносить получаемый уровень качества с трудоемкостью его реализации и возможностью достижения этой степени доверия.

Не все классы и семейства настоящего стандарта включены в каждый из перечисленных ниже оценочных уровней доверия. Эти семейства рекомендуется использовать для повышения уровней доверия в тех ПЗ и ЗБ, для которых они действительно полезны и необходимы. Определены семь иерархически упорядоченных оценочных уровней доверия для ранжирования при выборе доверия к безопасности объектов оценки. (Они, в некоторой степени, подобны пяти уровням зрелости технологий в стандарте **ISO 15504** — см. [4]). Каждый последующий ОУД представляет более высокое доверие (гарантированное качество), чем любой из предыдущих (ср. примеры в таблицах 1 и 2). Связь уровней доверия с классами и семействами технологических процессов обеспечения ЖЦ безопасности систем и программных средств в стандарте иллюстрированы семью таблицами. В каждой из них выделены и отмечены обязательные (белые) и рекомендуемые (остальные) классы и семейства процессов, обеспечивающие определенный уровень доверия при создании и применении методов и средств соответствующей безопасности. Эти таблицы помогают выбирать технологии в соответствии с требованиями к безопасности системы и ПС с учетом доступных ресурсов.

**Оценочный уровень доверия 1 (ОУД1)** — предусматривает функциональное тестирование и применим, когда требуется некоторая уверенность в правильном функционировании системы, а угрозы безопасности не рассматривают как серьезные. Он полезен там, где требуется, чтобы было уделено должное внимание безопасности, путем независимого тестирования на соответствие спецификации и экспертизы представленной документации. Предполагается, что оценка может успешно проводиться без помощи разработчика и с минимальными затратами, посредством анализа экспертами заданных функций безопасности с использованием функциональной спецификации, спецификации интерфейсов и руководств.

<b>Класс доверия 3</b>	<b>Компоненты доверия</b>
Управление конфигурацией	Средства контроля авторизации Охват УК объекта оценки
Поставка и эксплуатация	<b>Процедуры поставки</b> <b>Процедуры установки, генерации и запуска</b>
Разработка	<b>Неформальная функциональная спецификация</b> Детализация вопросов безопасности в проекте верхнего уровня <b>Неформальная демонстрация соответствия</b>
Руководства	<b>Руководство администратора</b> <b>Руководство пользователя</b>
Поддержка жизненного цикла	Идентификация мер безопасности
Тестирование	Анализ покрытия Тестирование: проект верхнего уровня <b>Функциональное тестирование</b> <b>Выборочное независимое тестирование</b>
Оценка уязвимостей	Экспертиза руководств <b>Оценка стойкости функции безопасности</b> <b>Анализ уязвимостей разработчиком</b>

Табл. 1

**Оценочный уровень доверия 2 (ОУД2)** — включает структурное тестирование, содержит требование сотрудничества с разработчиком для получения информации о проекте и результатах тестирования. Он применим в тех случаях, когда разработчикам или пользователям требуется независимо подтверждаемый уровень доверия (от невысокого до умеренного), при отсутствии доступа к полной документации при разработке. Такая ситуация может возникать при обеспечении безопасности разработанных ранее (наследуемых) систем или при ограниченной доступности к ним разработчика. ОУД2 обеспечивает доверие посредством анализа применяемых функций безопасности с использованием функциональной спецификации, спецификации интерфейсов, руководств и проекта верхнего уровня. Этот уровень требует тестирования и анализа уязвимостей разработчиком, основанного на более детализированных спецификациях.

**Оценочный уровень доверия 3 (ОУД3)** — таблица 1) — предусматривает методическое тестирование и проверку, позволяет разработчику достичь доверия путем применения проектирования безопасности без значительного изменения существующей технологии качественной разработки всей системы. ОУД3 применим в тех случаях, когда разработчикам или пользователям требуется независимо подтверждаемый умеренный уровень доверия на основе исследования системы и процесса ее разработки без существенных затрат на изменение технологии. Этот уровень представляет значимое увеличение доверия, требуя более полного покрытия тестированием функций и процедур безопасности.

**Оценочный уровень доверия 4 (ОУД4)** — предусматривает методическое проектирование, тестирование и углубленную проверку, что позволяет разработчику достичь максимального качества, основанного на регламентированной технологии разработки, которая не требует глубоких специальных знаний, навыков и других ресурсов. ОУД4 — самый высокий уровень, на который, вероятно, экономически целесообразно ориентироваться при оценке уже существующих продуктов. Анализ поддержан независимым тестированием, свидетельством разработчика об испытаниях, основанных на функциональной спецификации и проекте верхнего уровня, подтверждением результатов тестирования разработчиком и независимым анализом уязвимостей. Уровень также обеспечивает доверие посредством использования контроля среды разработки и дополнительного управления конфигурацией системы и ПС.

**Оценочный уровень доверия 5 (ОУД5)** — позволяет разработчику достичь максимального качества путем систематического проектирования безопасности, основанного на строгой технологии разработки, поддержанной умеренным применением узко специализированных методов, не влекущих излишних затрат на методы проектирования безопасности. Доверие достигается применением формальной модели политики безопасности и полуформального представления функциональной спецификации и проекта верхнего уровня системы, а также полуформальной демонстрации соответствия между ними. Кроме этого, требуется модульное проектирование системы. Анализ поддержан независимым свидетельством разработчика об испыта-

Класс доверия 7	Компоненты доверия
Управление конфигурацией	<b>Полная автоматизация УК</b> <b>Расширенная поддержка</b> <b>Охват УК инструментальных средств разработки</b>
Поставка и эксплуатация	Предотвращение модификации <b>Процедуры установки, генерации и запуска</b>
Разработка	Формальная функциональная спецификация Формальный проект верхнего уровня <b>Структурированная реализация функциональной безопасности</b> Минимизация сложности <b>Полуформальный проект нижнего уровня</b> Формальная демонстрация соответствия <b>Формальная модель политики безопасности</b>
Руководства	<b>Руководство администратора</b> <b>Руководство пользователя</b>
Поддержка жизненного цикла	<b>Достаточность мер безопасности</b> Измеримая модель жизненного цикла <b>Соответствие всех частей объекта оценки стандартам реализации</b>
Тестирование	<b>Строгий анализ покрытия</b> Тестирование на уровне реализации <b>Упорядоченное функциональное тестирование</b> Полное независимое тестирование
Оценка уязвимостей	<b>Систематический анализ скрытых каналов</b> <b>Анализ и тестирование опасных состояний</b> <b>Оценка стойкости функции безопасности</b> <b>Высокостойкий</b>

Табл. 2

ниях, основанных на функциональной спецификации, проектах верхнего и нижнего уровня, независимым подтверждением результатов тестирования разработчиком и независимым анализом уязвимостей. ОУД5 также обеспечивает качество посредством использования контроля среды разработки и управления конфигурацией системы, требуя соблюдать структурированную архитектуру системы.

**Оценочный уровень доверия 6 (ОУД6)** — позволяет разработчикам достичь высокой безопасности путем полуформальной верификации всего проекта и тестирования, применением специальных методов проектирования безопасности в строго контролируемой среде разработки с целью получения высокой безопасности системы и защиты активов от значительных рисков, где ценность защищаемых активов оправдывает дополнительные затраты. ОУД6 также обеспечивает повышение доверия посредством использования структурированного процесса разработки, контроля среды разработки и управления конфигурацией системы, включая полную автоматизацию, и свидетельства безопасности процедур поставки. Этот уровень представляет значительное увеличение доверия по сравнению с предыдущим, требует всестороннего анализа, структурированное представление реализации, более стройную структуру системы, всесто-

ронный независимый анализ уязвимостей, систематическую идентификацию скрытых каналов, улучшенное управление конфигурацией и глубокий контроль среды разработки.

**Оценочный уровень доверия 7 (ОУД7)** — таблица 2) -применим при разработке безопасных систем для использования в ситуациях чрезвычайно высокого риска и/или там, где высокая ценность активов или систем оправдывает максимальные затраты на их безопасность. Практическое применение уровня ограничено системами, которые строго ориентированы на реализацию полных функциональных возможностей безопасности, для которых возможен и целесообразен подробный формальный анализ. Уровень обеспечивает качество посредством использования всех представленных выше классов и семейств, а также процессов предшествующих уровней, структурированного процесса разработки, средств контроля среды разработки и всестороннего управления конфигурацией системы, включая полную автоматизацию, и свидетельства безопасных процедур поставки (см. рис. 2). Этот уровень представляет значительное увеличение доверия, требует всестороннего анализа, использующего формальные представления и формальное соответствие, а также всестороннее независимое тестирование.

### 3. Особенности методологии обеспечения безопасности программных средств в стандарте ISO 13335

Широкий комплекс методологических задач, которые необходимо решать при проектировании безопасности систем, отражает стандарт **ISO 13335:1996-1998 – 1-5**. Руководство по управлению безопасностью. В его пяти частях 1. Концепция и модели обеспечения безопасности информационных технологий; 2. Планирование и управление безопасностью информационных технологий; 3. Техника управления безопасностью ИТ; 4. Селекция (выбор) средств обеспечения безопасности; 5. Безопасность внешних связей внимание сосредоточено на основных принципах и методах проектирования, равнопрочных систем обеспечения безопасности информационных систем от угроз различных видов. Это руководство достаточно полно систематизирует основные методы и процессы подготовки проекта защиты для последующей разработки конкретной комплексной системы обеспечения безопасности функционирования ИС. Изложение базируется на понятии риска от угроз любых негативных воздействий на ИС. В первой части стандарта представлены функции средств защиты и необходимые действия по их реализации, модели уязвимости и взаимодействие средств защиты. При проектировании систем защиты, рекомендуется учитывать: необходимые функции защиты; угрозы; уязвимость; негативные воздействия; риски; защитные меры; ресурсы (аппаратные, информационные, программные, специалистов) и их ограниченность. В остальных частях стандарта предложена и развивается концепция и модель управления и планирования построения системы обеспечения безопасности, упрощенная схема компонентов которой представлена на рис. 3.

В стандарте выделены функциональные компоненты и средства обеспечения безопасности, а также их взаимодействие. Процессы управления безопасностью должны включать:

- управление изменениями и конфигурацией функций и компонентов системы безопасности;
- анализ и управление рисками;
- прослеживаемость функций и результатов комплексов средств обеспечения безопасности;
- регистрацию, обработку и мониторинг инцидентов.

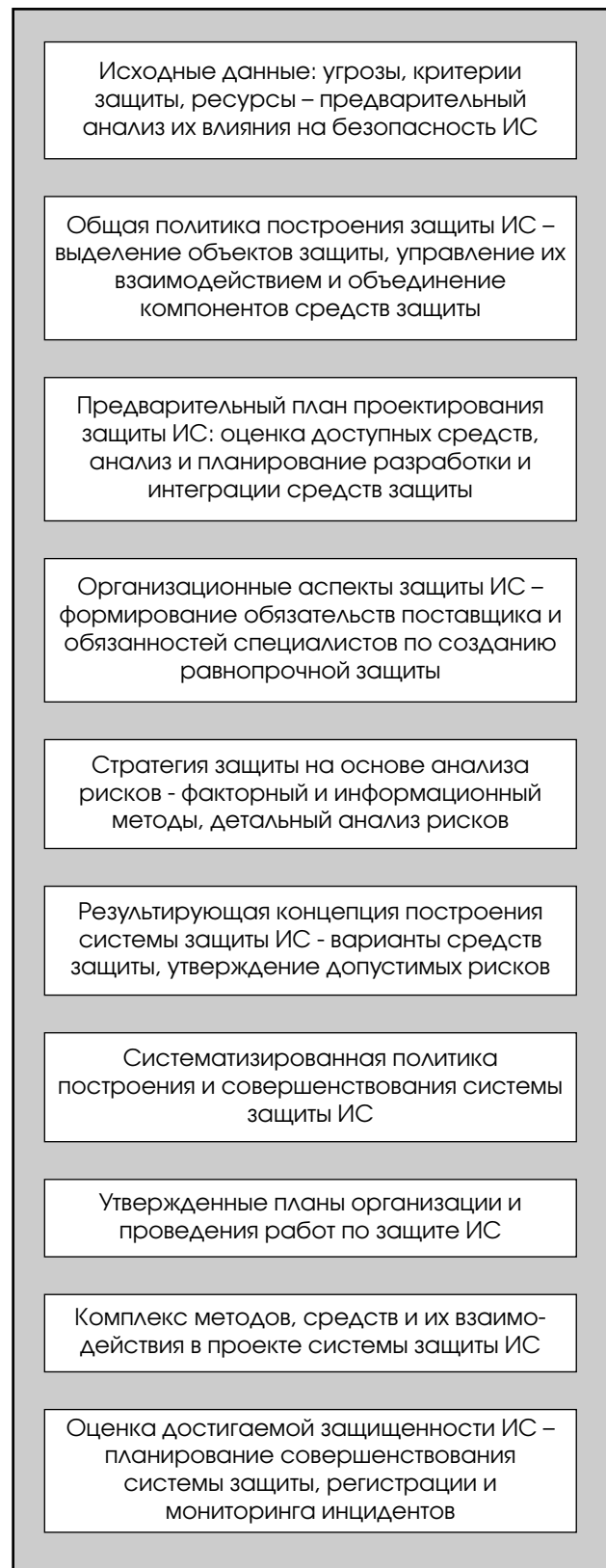


Рис. 3.

Приводятся общие требования к оценке результатов обеспечения безопасности, а также возможные варианты организации специалистов для



комплексного обеспечения безопасности ИС. Систематизирована политика и техника планирования, выбора, построения и использования средств обеспечения безопасности для ограничения допустимых рисков при различных схемах взаимодействия и средствах защиты. Рекомендуются различные подходы и стратегии при создании систем защиты и поддержке их последующего развития. Содержание частей стандарта детализирует общие концепции и достаточно точно определяется их названиями. Изложенную в стандарте модель планирования обеспечения безопасности, целесообразно конкретизировать и использовать как **фрагмент проекта функциональной безопасности ПС**.

## 4. Особенности процессов жизненного цикла программных средств в стандарте ISO 12207

Стандарт ISO 12207:1995 – Процессы жизненного цикла программных средств – наиболее полно на уровне международных стандартов отражает **жизненный цикл, технологию разработки и обеспечения качества сложных программных средств**. Жизненный цикл ПС представлен набором этапов, частных работ и операций в последовательности их выполнения и взаимосвязи, регламентирующих ведение разработки на всех стадиях от подготовки технического задания до завершения испытаний ряда версий и окончания эксплуатации ПС. В ЖЦ включаются описания исходной информации, способов выполнения операций и работ, устанавливаются требования к результатам и правилам их контроля, а также к содержанию технологических и эксплуатационных документов. Определяется организационная структура коллективов, распределение и планирование работ, а также контроль за реализацией ЖЦ ПС.

Стандарт может использоваться как непосредственный директивный, руководящий или ре-

комендательный документ, а также как организационная база при создании средств автоматизации соответствующих технологических этапов или процессов. Для реализации положений стандарта должны быть выбраны инструментальные средства, совместно образующие взаимосвязанный комплекс технологической поддержки и автоматизации ЖЦ и не противоречащие предварительно скомпонованному набору нормативных документов. Имеющиеся в стандарте пробелы следует заполнять спецификациями или нормативными документами, регламентирующими применение выбранных или созданных инструментальных средств автоматизации разработки и документирования ПС.

Стандарт определяет архитектуру, процессы, разделы и подразделы ЖЦ ПС, а также перечень базовых работ и детализирует содержание каждой из них. Архитектура ЖЦ ПС в стандарте базируется на **трех крупных компонентах** (рис. 4):

- основные процессы жизненного цикла ПС и определяющие работы (раздел 5);
- вспомогательные процессы и работы, поддерживающие жизненный цикл ПС (раздел 6);
- организационные процессы и управление жизненным циклом ПС (раздел 7).

Эти разделы стандарта состоят из ряда подразделов, в которых подробно раскрывается содержание каждой работы и комментируются особенности их выполнения. Рекомендации к каждому подразделу состоят в среднем из 3-6 пунктов – работ (процедур). Общее число работ и комментариев к ним в стандарте свыше 220.

В **разделе 5** изложены основы ЖЦ и рекомендации по подготовке, разработке, эксплуатации и сопровождению программных средств (см. рис.4). Процессы приобретения и/или подготовки к созданию ПС должны начинаться с инициализации проекта, анализа концепции, анализа рынка продуктов, выработки требований и состава поддерживающих документов, создания предварительного плана проекта. Основные работы по созданию сложного комплекса программ рекомендуется начинать с определения состава сопровождающих документов, выбора средств конфигурационного управления и обеспечения качества, а также выбора методов и средств технологического обеспечения разработки всей информационной системы. Кодирование и тестирование каждого компонента ПС должно быть оформлено совокупностью документов, удостоверяющих соответствие компонента первичной спецификации, содержащих тесты и результаты тестирования.

Рекомендуется разрабатывать план работ, включающий комплексирование компонентов, тестирование по всем разделам требований и показа-

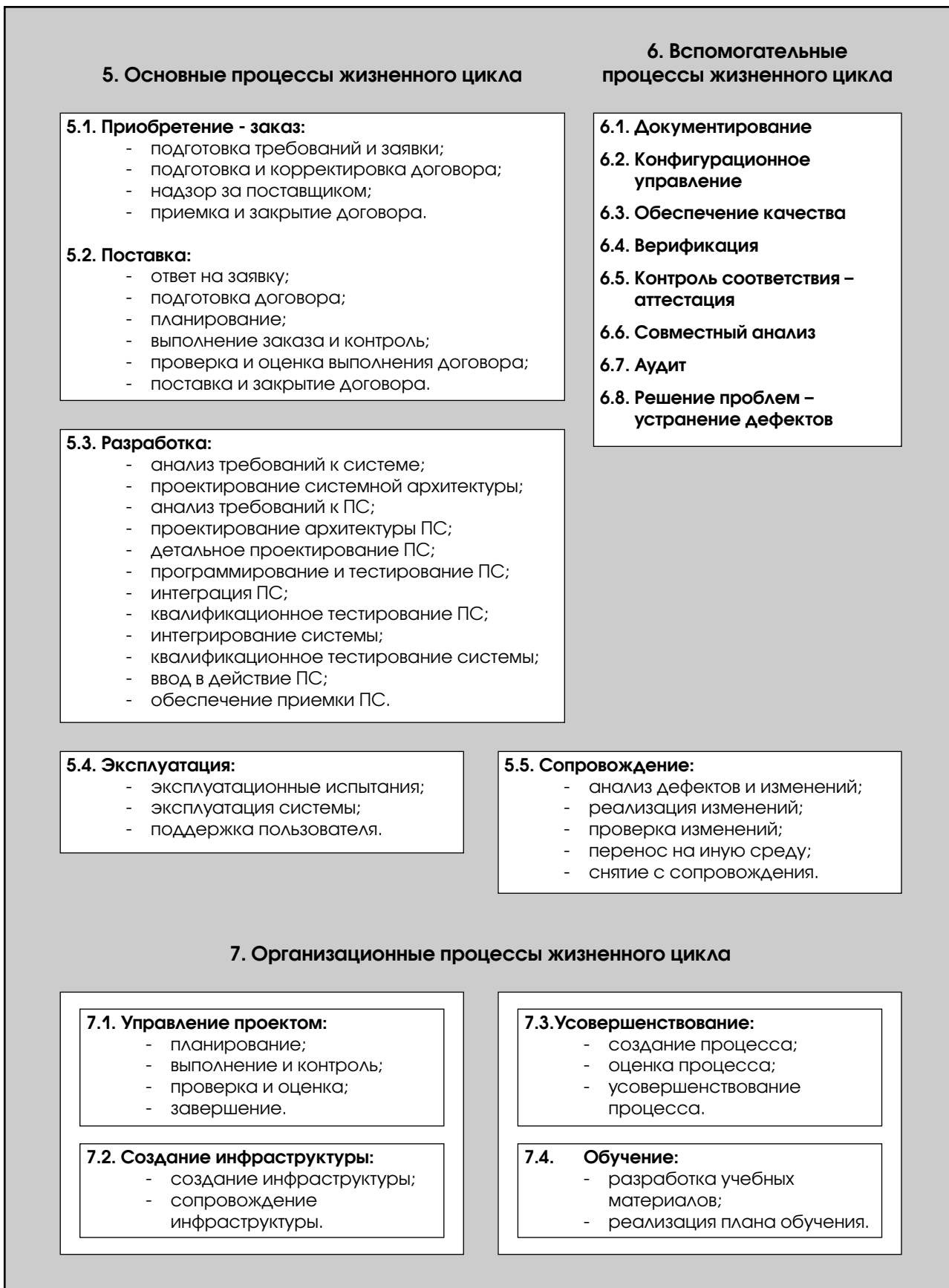


Рис. 4.

телям качества, а также документирование плана, результатов интеграции, использованных тестов, критериев оценки и полученных результатов. Далее ПС следует подвергать квалификационному (аттестационному) тестированию по всем разделам требований контракта, при широком варьировании тестов, изменениях значений критериев, а также тестировать полноту и адекватность технологической и пользовательской документации реальному программному продукту. Проверенный таким образом комплекс программ интегрируется в вычислительные средства информационной системы, средства визуализации и телекоммуникации.

Эти работы взаимодействуют с работами, обеспечивающими сопровождение ПС. Специалисты анализируют сообщения об ошибках и предложения на модификацию ПС, селективируют их на соответствие требованиям контракта и оценивают целесообразность проведения изменений. Подготовленные изменения тестируются и проверяются по критериям, определенным в документации.

Вспомогательные технологические работы, поддерживающие жизненный цикл ПС, и рекомендации по их выполнению изложены в **разделе 6**. Процессы документирования ПС должны охватывать планирование и обеспечение документирования, рекомендации по стандартизации, проектированию и разработке, а также по производству, конфигурационному управлению и сопровождению комплекта документации на ПС. Конфигурационное управление предлагается включать в общий план управления проектом. Для обеспечения гарантий качества следует использовать планирование, методологию, процедуры и стандарты поддержки качества ПС в соответствии с контрактом с учетом доступных ресурсов. Верификация ПС должна включать ее организацию, планирование и техническое обеспечение. Удостоверение правильности (аттестация) должна гарантировать полное соответствие программного продукта спецификациям, требованиям и документации на ПС и возможность его надежного функционирования и безопасного применения пользователем.

Управление проектом должно быть сосредоточено, в основном, в подготовке и обеспечении планирования и управления ресурсами, персоналом, аппаратурой, программными средствами и инструментарием. Процессы ревизии – аудита служат для установления соответствия реальных работ и отчетов требованиям, планам и контракту. В процессе решения задач должны выявляться и регистрироваться проблемы и дефекты последующего применения программных средств и их функционирования.

Организации жизненного цикла ПС посвящен **раздел 7**. Она включает основные работы по уп-

равлению проектом, производством и средствами для обеспечения процессов по разработке, эксплуатации и сопровождению. Процессы формирования инфраструктуры должны состоять из выбора и установления аппаратных и программных средств, технологии, стандартов и обслуживания, используемых для разработки, сопровождения и обеспечения эксплуатации ПС. Процессы совершенствования жизненного цикла ПС состоят в установлении, оценивании, измерении, контроле и корректировке процессов жизненного цикла конкретных ПС. Процессы обучения определяются требованиями к проекту, должны учитывать необходимые ресурсы, управление и технические средства. Изложены рекомендации по преобразованию и адаптации базовой структуры этого международного стандарта для конкретного проекта (приложение А) и руководство по их выполнению в ЖЦ ПС (приложение В).

Стандартом **ISO 15271:1998** – Руководство по применению **ISO 12207** – поддержано **практическое использование этого стандарта**. Он содержит подробные рекомендации по внедрению, применению в проектах ПС, а также при организации работ и реализации требований стандарта **ISO 12207**.

Стандартом **ISO 16326:1999** Руководство по применению **ISO 12207** при административном управлении проектами регламентированы **процессы управления проектированием**. Детально изложены работы по планированию и процедуры выполнения процесса административного управления на различных этапах жизненного цикла ПС.

## 5. Особенности процессов жизненного цикла программных средств в стандарте ISO 15504

Стандарт **ISO 15504:1-9:1998** – Оценка (аттестация) процессов жизненного цикла программных средств – предоставляет базу для реализации на предприятиях и в проектах процессов жизненного

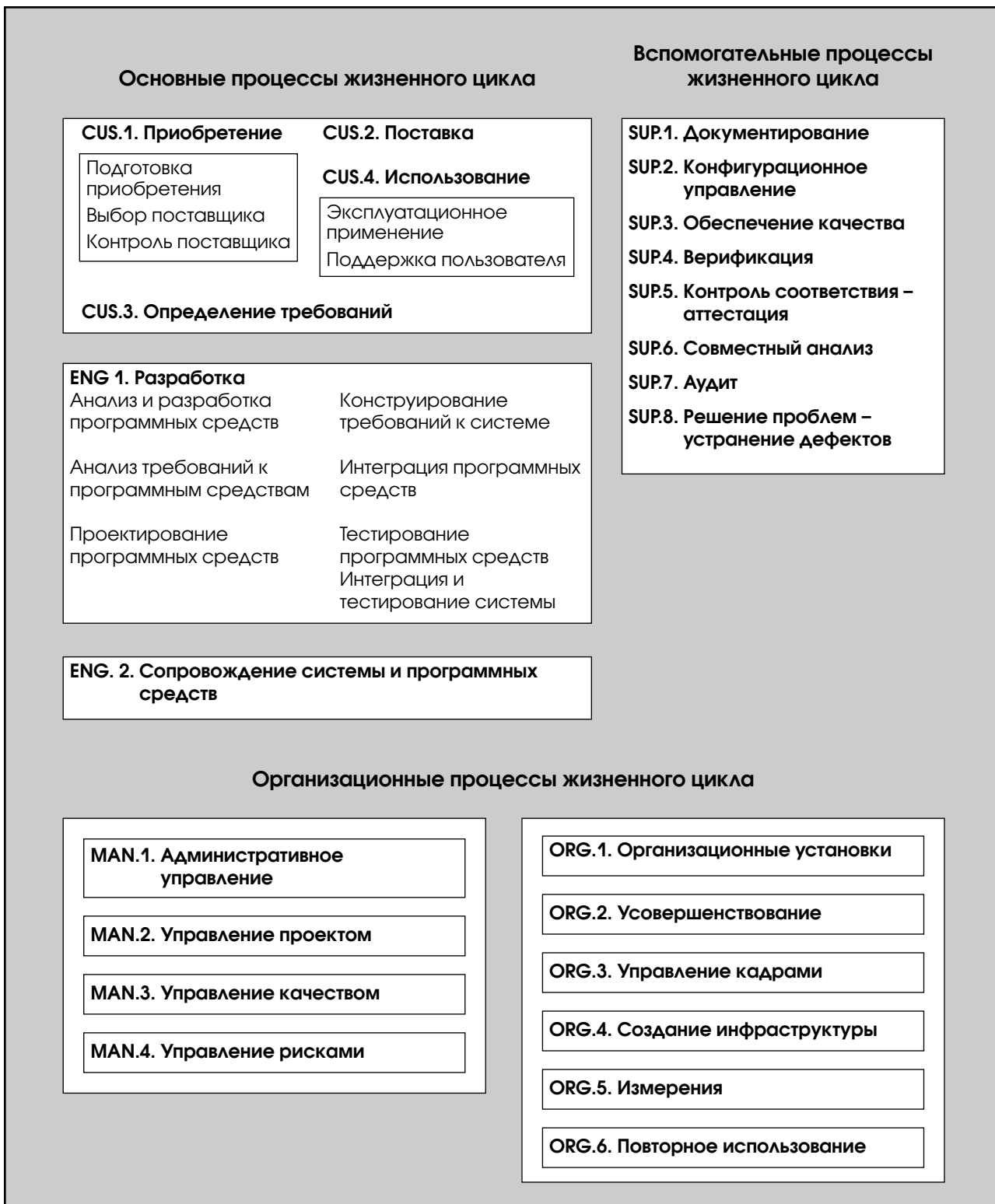


Рис. 5.

цикла ПС, регламентированных стандартом **ISO 12207**. Рубрикации основных процессов в этих двух стандартах подобны (рис. 4 и рис. 5). В стандарте **ISO 15504** модернизирован и несколько расширен состав организационных процессов, и более по-

дробно детализированы работы во всех стандартизированных процессах жизненного цикла ПС. Поэтому оба стандарта целесообразно применять совместно при конкретизации жизненного цикла реальных проектов сложных комплексов программ.

Аттестация реализации ЖЦ ПС направлена на обеспечение адекватности управления процессами и должна учитывать внешнюю среду, в которой выполняются аттестуемые процессы. Чтобы предприятие могло улучшить качество своей продукции, оно должно иметь проверенный, последовательный и надежный метод для аттестации состояния своих технологических процессов, а также иметь инструментальные средства использования ее результатов как часть Программы усовершенствования процессов ЖЦ ПС. Использование аттестации процессов внутри предприятия должно способствовать выработке культуры постоянного совершенствования и повышения характеристик качества в ЖЦ ПС, а также соответствующих механизмов поддержания этой культуры и оптимизации использования ресурсов. Это должно приводить к появлению зрелых организаций, обеспечивающих минимальную стоимость полного жизненного цикла своей продукции и, как результат, максимально удовлетворяющих требования конечного пользователя к характеристикам качества и безопасности ПС.

Покупателям и заказчикам ПС **выгодно использование аттестации** процессов ЖЦ при определении зрелости поставщика, что:

- уменьшит неопределенность при выборе поставщиков программных комплексов за счет того, что риски, связанные с реальной зрелостью подрядчика, выявляются еще до заключения договора;
- позволит заранее предусмотреть необходимые меры на случай возникновения рисков события;
- предоставит количественные критерии выбора при сопоставлении потребностей бизнеса, требований и оценочной стоимости проекта ПС с реальной зрелостью конкурирующих поставщиков;
- приведет к общему пониманию необходимости использования результатов аттестации для совершенствования процессов и оценки зрелости поставщика при прогнозировании характеристик ЖЦ ПС.

Для достижения устойчивых результатов в процессе развития технологии и организации управления жизненным циклом ПС в стандарте **ISO 15504** рекомендуется методология обеспечения качества сложных программных средств **СММ** (Capability Maturity Model) – **система и модель оценки зрелости** комплекса, применяемых технологических процессов [4]. Модель основана на формализации и использовании пяти уровней зрелости технологий поддержки ЖЦ ПС, которые оп-

ределяют потенциально возможное качество и безопасность создаваемых комплексов программ. Эти **уровни зрелости характеризуются** степенью формализации, адекватностью измерения и документирования процессов и продуктов ЖЦ ПС, широтой применения стандартов и инструментальных средств автоматизации работ, наличием и полнотой реализации функций системой обеспечения качества технологических процессов и их результатов. Они, в некоторой степени, подобны семи оценочным уровням доверия в стандарте **ISO 15408-3**.

**Уровень 1 Начальный.** Наиболее массовые разработки проектов ПС характеризуются относительно небольшими объемами программ в несколько тысяч строк, создаваемых несколькими специалистами. Они применяют простейшие не формализованные технологии с использованием типовых инструментальных компонентов операционных систем. Основные процессы ЖЦ ПС на этом уровне не регламентированы, выполняются не совсем упорядоченно и зависят от некоординированных индивидуальных усилий специалистов. Успех проекта, как правило, зависит от энергичности, таланта и опыта нескольких руководителей и исполнителей. Процессы на первом уровне характеризуются своей непредсказуемостью по срокам в связи с тем, что их состав, назначение и последовательность выполнения могут меняться случайным образом в зависимости от текущей ситуации.

**Уровень 2 Управляемый.** Для сложных проектов ПС объемом в десятки и сотни тысяч строк, в которых участвуют десятки специалистов разной квалификации, необходимы организация, регламентирование технологии и унификация процессов деятельности каждого из них. Процессы на этом уровне заранее планируются, их выполнение контролируется, чем достигается предсказуемость результатов и времени выполнения этапов, компонентов и проекта в целом. Основной особенностью второго уровня является наличие формализованных и документированных процессов управления проектами, которые пригодны для модернизации, а их результаты, поддаются количественной оценке. На этом уровне акценты управления сосредотачиваются на предварительном упорядочении и регламентировании процессов создания, сопровождения и оценивания качества программного средства, однако для крупномасштабных проектов ПС с гарантированным качеством, риск провала остается еще достаточно большим.

**Уровень 3 Определенный.** При высоких требованиях заказчика и пользователей к конкретным характеристикам качества сложного ПС и к выполнению ограничений по использованию ресурсов, необходимо дальнейшее совершенствование и повышение уровня зрелости процессов ЖЦ ПС. Процессы ЖЦ ПС на этом уровне должны быть стандартизированы, и представлять собой единую технологическую систему, обязательную для всех подразделений. На основе единой технологии поддержки и обеспечения качества ЖЦ ПС, для каждого проекта могут разрабатываться дополнительные процессы последовательного оценивания качества продуктов с учетом их особенностей. Описание каждого процесса должно включать условия его выполнения, входные данные, рекомендации стандартов и процедуры выполнения, механизмы проверки качества результатов, выходные данные, условия и документы завершения процессов. В описания процессов включаются сведения об инструментальных средствах, необходимых для их выполнения, роль, ответственность и квалификация специалистов.

**Уровень 4 Предсказуемый.** Для реализации проектов крупномасштабных, особенно сложных ПС в жестко ограниченные сроки и с высоким гарантированным качеством, необходимы активные меры для предотвращения и выявления дефектов и ошибок на всех этапах ЖЦ ПС. Управление должно обеспечивать выполнение процессов в соответствии с текущими требованиями к характеристикам качества компонентов и ПС в целом. На этом уровне должна применяться система детального поэтапного оценивания характеристик качества, как технологических процессов ЖЦ, так и самого создаваемого программного продукта и его компонентов. Должны разрабатываться и применяться универсальные методики количественной оценки реализации процессов и их качества. Одновременно с повышением сложности и требований к качеству ПС, следует совершенствовать управление проектами за счет сокращения текущих корректировок и исправлений дефектов при выполнении процессов. Результаты процессов становятся предсказуемыми по срокам и качеству в связи с тем, что они измеряются в ходе их выполнения и реализуются в рамках заданных ресурсных ограничений.

**Уровень 5 Оптимизируемый.** Дальнейшее последовательное совершенствование и модернизация технологических процессов ЖЦ ПС для повышения качества их выполнения и расширение глубины контроля за их реализацией. Одна из основных целей этого уровня сокращение проявле-

ний и потерь от случайных дефектов и ошибок путем выявления сильных и слабых сторон используемых процессов. При этом приоритетным является анализ рисков, дефектов и отклонений от заданных требований заказчика. Эти данные также используются для снижения себестоимости ЖЦ особо сложных ПС в результате внедрения новых технологий и инструментария, а также для планирования и осуществления модернизации всех видов процессов. Технологические нововведения, которые могут принести наибольшую выгоду, должны стандартизироваться и адаптироваться в комплексную технологию обеспечения и оценивания системы качества предприятия и его продукции.

Виды деятельности для высоких уровней зрелости в соответствии с СММ, в стандарте делятся на базовые и общие. Базовые виды деятельности являются обязательными и сгруппированы в пять категорий (см. рис. 5).

**Контрактная** категория (заказчик-поставщик) (CUS) состоит из видов деятельности, непосредственно влияющих на взаимодействие с заказчиком, они поддерживают процессы организации разработки, испытаний и передачи ПС заказчику и обеспечивают возможность его корректного использования.

**Инженерная** категория (ENG) включает виды деятельности, которые непосредственно определяют, реализуют или поддерживают программный продукт и документацию на него.

**Управленческая** категория (MAN) определяет все аспекты управления проектом и координацию использования его ресурсов в ЖЦ или при предоставлении услуг, удовлетворяющих заказчиков ПС.

**Вспомогательная** категория (SUP) виды деятельности, которые обеспечивают реализацию и совершенствование основных процессов, а также поддерживают производительность и качество процессов в проекте.

**Организационная** категория (ORG) определяет цели предприятия-разработчика и формирует методы управления, необходимые для повышения качества использования ресурсов и всего ЖЦ ПС.

Девять частей стандарта **ISO 15504**, посвящены различным базовым задачам, относящимся к оцениванию, аттестации и совершенствованию зрелости процессов ЖЦ ПС на предприятии. Стандарт **ISO 15504** связан с другими международными стандартами, он дополняет некоторые стандарты и другие модели для оценки зрелости, качества и эффективности предприятий и процессов ЖЦ ПС. Этот стандарт преследует ту же цель, что и серия стандартов **ISO 9000:2000** — формализации и обеспечения уверенности в достаточности системы уп-

равления качеством продукции у поставщика. Одновременно предоставляется потребителям основа для оценки того, обладают ли потенциальные поставщики производственными возможностями, отвечающими потребностям заказчиков. Аттестация процессов дает пользователям возможность оценить зрелость процессов обеспечения ЖЦ ПС по непрерывной шкале таким образом, что эти оценки сопоставимы и повторяемы.

Стандарты **ISO 12207** и **ISO 15504** дополнительно поддерживаются группой стандартов, детализирующих отдельные этапы и процессы жизненного цикла, которые целесообразно применять для обеспечения функциональной безопасности и высокого качества сложных программных средств [3, 6]:

- **ISO 12182:1998.** ИТ. Классификация программных средств.
- **ISO 9126:1991.** ИТ. Оценка программного продукта. Характеристики качества и руководство по их применению.
- **ISO 14598-1-6:1998-2000.** Оценивание программного продукта. Ч.1. Общий обзор. Ч. 2. Планирование и управление. Ч. 3. Процессы для разработчиков. Ч.4. Процессы для покупателей. Ч.5. Процессы для оценщиков. Ч. 6. Документирование и оценивание модулей.
- **ISO 14756: 1999.** ИТ. Измерение и оценивание производительности программных средств компьютерных вычислительных систем.
- **ISO 12119:1994.** ИТ. Требования к качеству и тестирование.
- **ISO 15846:1998.** ТО. Процессы жизненного цикла программных средств. Конфигурационное управление программными средствами.
- **ISO 14764: 1999.** ИТ. Сопровождение программных средств.
- **ISO 15910:1999.** ИТ. Пользовательская документация программных средств.
- **ISO 6592:2000.** ОИ. Руководство по документации для вычислительных систем.
- **ISO 9294:1990.** ТО. ИТ. Руководство по управлению документированием программного обеспечения.

## 6. Особенности процессов разработки и документирования встроенных программных средств в стандарте ГОСТ Р 51904

Стандарт **ГОСТ Р 51904-2002** Программное обеспечение встроенных систем. Общие требования к разработке и документированию создан в развитие **ISO 12207** с целью учета специфики жизненного цикла программных средств встроенных систем реального времени высокого качества, преимущественно для авиационных, космических и транспортных систем. В стандарте значительное внимание уделяется обеспечению качества и функциональной безопасности ПС, чем структура и рекомендации этого стандарта наиболее близки к **IEC 61508**. В стандарте представлены: общие требования (п. 4), системные аспекты, связанные с разработкой ПС (п.5), и шесть крупных разделов (п.п. 6 – 11), подробно описывающие и рекомендуемые основные процессы ЖЦ встроенных ПС. Последовательно, детально рассмотрены требования и методы реализации процессов жизненного цикла сложных ПС. Выделена группа **процессов планирования**, которые определяют и координируют для проекта действия процессов разработки и интегральных процессов. **Процессы разработки**, в ходе выполнения которых создается программное средство, отражены в разделах:

- определение требований к ПС: состав работ, выполняемых при определении требований к ПС; установление модели жизненного цикла ПС; критерии переходов между процессами; общие требования для разработки; стандарты; ПС многократного использования; требования к системе; отработка критических требований;
- планирование: состав работ, выполняемых в процессе планирования ПС; планирование среды жизненного цикла ПС; язык программирования и компилятор; стандарты разработки; состав работ, выполняемых в процессе кодирования программ;
- проектирование и разработка ПС: состав работ, выполняемых в процессе проектирования ПС; потоки информации между процессами жизненного цикла системы и ПС; отказовые ситуации и назначение уровня ПС; анализ системных требований; анализ информации о потребностях пользователя; проектирование архитектуры системы; мониторинг функцио-

нальной безопасности ПС; подготовка руководств пользователя; интеграция компонентов.

**Интегральные процессы**, которые обеспечивают корректную реализацию и качество выполнения процессов разработки и их выходных данных детально представлены в разделах:

- верификация ПС: состав работ, выполняемых в процессе верификации ПС; просмотры и анализы требований верхнего уровня; архитектуры ПС; требований нижнего уровня; исходного кода; тестовых вариантов, процедур и результатов; выбор тестовых вариантов, основанных на требованиях; анализ тестового покрытия; интеграционное тестирование; квалификационное тестирование системы;
- управление конфигурацией ПС: состав работ, выполняемых в процессе управления конфигурацией ПС; отчетность о дефектах, трассируемость и корректирующие действия; архивирование и получение документов; выпуск версий; контроль среды жизненного цикла ПС; аудит конфигурации;
- обеспечение качества ПС: состав работ, выполняемых в процессе обеспечения качества ПС; просмотр согласованности; документирование обеспечения качества ПС; независимость в обеспечении качества ПС;
- сертификационное сопровождение: средства согласования и планирования; обоснование согласованности; минимальный состав документов жизненного цикла ПС, передаваемых сертифицирующей организации; документы жизненного цикла ПС, относящиеся к типовому проекту; критерии и документация по аттестации для инструментальных средств разработки и верификации ПС.

Интегральные процессы должны выполняться частично одновременно с процессами разработки ПС. В рамках конкретного проекта должны быть установлены одна или несколько моделей жизненного цикла ПС, в соответствии, с которыми выбираются необходимые процедуры для каждого процесса, определяется последовательность их выполнения, назначаются ответственные за выполнение работ. Для конкретного проекта последовательность процессов определяется сложностью проекта в целом, функциональными возможностями разрабатываемой системы, объемом и сложностью ПС, стабильностью требований, использованием ранее полученных результатов, стратегией разработки и возможностями аппаратных средств.

Для всех работ по созданию ПС должны использоваться систематизированные, зарегистриро-

ванные методы. План разработки ПС должен содержать описание этих методов или включать в себя ссылки на источники и стандарты, в которых они описаны. Следует разработать и использовать руководства для представления требований: проекта, кода, тестовых вариантов, тестовых процедур и результатов тестирования. В документе представлены подробные рекомендации, на которых акцентируется дополнительное внимание и детализируются положения стандарта **ISO 12207. Основные требования и рекомендации стандарта сводятся к следующим.**

Разработчик должен принимать участие в анализе, определении и документировании требований, которым должна удовлетворять встроенная система и ПС, и методы, которые необходимо использовать в целях гарантирования выполнения каждого требования. Эта информация может быть представлена в форме предложений, обзоров, сообщений о дефектах и изменениях, обратной связи к прототипам, интервью о потребностях пользователя или в любой другой форме.

Если системные требования предусматривают возможность модификации, осуществляемой пользователем, то они могут изменять ПС в заданном диапазоне без рассмотрения, осуществляемого сертифицирующей организацией. В этом случае системные требования должны определить механизмы, которые устраняют негативное влияние на безопасность ПС модификации, осуществляемой пользователем, независимо от того, как она выполнена. При проведении модификации пользователем последний должен нести ответственность за все аспекты модифицируемого им ПС, например, за управление конфигурацией, обеспечение безопасности и качества и верификацию.

Рекомендуется идентифицировать компоненты или части их, критические с точки зрения безопасности, сбой в которых может привести к отказовой ситуации. Если имеется такое ПС или компонент, следует предусмотреть стратегию обеспечения его защиты. Стратегия должна гарантировать методы, при которых требования, проект, реализация и эксплуатационные процедуры для ПС, минимизируют или устраняют потенциальные нарушения безопасности ПС. Следует проанализировать требования контракта, относящиеся к использованию ресурсов аппаратных средств компьютера (например, максимально возможная производительность процессора, объем памяти, пропускная способность устройств ввода/вывода).

Разработчик должен принимать участие в определении и документировании проектных решений системного уровня. Эти решения являются прерогативой разработчика, если они формально



не преобразованы в требования при выполнении контракта. Он ответствен за выполнение всех требований и демонстрацию этого выполнения посредством квалификационного тестирования. Реализация проектных решений, действующих как внутренние требования, должна быть подтверждена внутренним тестированием разработчика, выполнение которого нет необходимости демонстрировать заказчику.

Рекомендуется участие разработчика в определении и документировании проекта архитектуры ПС (идентификации компонентов, их интерфейсов и концепции их совместного выполнения) и в прослеживании соответствия между компонентами ПС и системными требованиями. В процессе оценки безопасности должно устанавливаться, как архитектурное проектирование ПС предотвращает аномальное поведение при появлении отказовых ситуаций. Должны применяться архитектурные стратегии, которые позволяют ограничивать воздействие дефектов, обнаруживать ошибки и обеспечивать приемлемую реакцию ПС для устранения их воздействия. Библиотека разработки ПС может быть частью среды разработки и среды верификации. Следует сопровождать Библиотеку разработки ПС на протяжении действия контракта.

Разработчик должен подготовить исполняемое ПС для передачи в организацию, осуществляющую сопровождение, а также файлы, необходимые для установки и эксплуатации ПС на объектной ЭВМ. Он должен принимать участие в совместных с заказчиком технических просмотрах, проводимых в течение всего периода выполнения контракта. В этих просмотрах, как со стороны разработчика, так и со стороны заказчика должны принимать участие лица с достаточными техническими знаниями о разрабатываемом ПС.

Следует осуществлять контроль за критическими для выполнения контракта ситуациями, которые могут возникнуть во время разработки ПС. Необходимо выявлять, идентифицировать и анализировать потенциальные технические, стоимостные или временные критические ситуации и риски; разработать стратегии для предотвращения или устранения таких ситуаций; регистрировать возможные риски и стратегии их предотвращения и реализовать эти стратегии в соответствии с Планом. В течение всего жизненного цикла ПС должны создаваться документы, чтобы планировать требуемые действия, управлять, объяснять, регистрировать выполнение требуемых действий. Эти документы должны отражать реализацию процессов жизненного цикла ПС, сертификацию системы и последующую модификацию ПС. Заказчик должен осуществлять выбор необходимого и экономически обос-

нованного состава и содержания документов для конкретной разработки из представленных в стандарте 39-и типов и структур. Их форма должна обеспечивать эффективный поиск и просмотр документов жизненного цикла ПС в процессе обслуживания системы. Состав документов и их конкретная форма должны быть определены в Плане документирования ПС. Заказчик на основании информации, полученной от разработчика, должен определить, какие руководства являются необходимыми для данной системы, и требовать разработки только этих руководств.

## 7. Особенности создания программных средств в системах безопасности атомных электростанций по стандарту МЭК 60880

Стандарт МЭК 60880 состоит из двух частей под общим названием — Программное обеспечение компьютеров в системах безопасности атомных электростанций. **Первая часть утверждена в 1986 году** и естественно несколько устарела. **Вторая часть, утвержденная в 2000 году**, является дополнением и развитием первой части по трем специальным направлениям, которые отражены в ее подзаголовке — Программные аспекты защиты от отказов по общим причинам; использование программных инструментов; применение ранее разработанного программного обеспечения. Основная особенность этой части стандарта состоит в концентрации рекомендаций на методах и процедурах, обеспечивающих и гарантирующих высокое качество и безопасность функционирования программ на всех этапах их жизненного цикла, предотвращающих ошибки и отказы системы. Почти половина объема в обеих частях стандарта уделена Приложениям, в которых конкретизируются и детализируются общие рекомендации.

В **первой части** рассмотрены общие вопросы и некоторые фазы жизненного цикла ПС, задачи обеспечения качества, верификация, испытания и документирование комплексов программ. Рекомендуется тщательно разрабатывать требования к ПС и подробно отражать в них: функции, конфигурацию системы, взаимодействие с внешней средой, ограничения характеристик аппаратуры и комплекса программ, необходимость непрерывного контроля программных и аппаратных средств, методы и процессы периодических и заключительных испытаний системы. Выделен раздел рекомендаций процессов разработки ПС – проектирования и программирования (кодирования) программ. Обращается внимание на необходимость самоконтроля логики и данных программ, на декомпозицию и модульное построение ПС, на стройность и простоту структуры программ для сокращения дефектов и ошибок. Для этого же предлагается подробная верификация программ по фазам ЖЦ ПС, планирование тестирования, проверка критических ситуаций, специфицирование испытаний и их документирование. В разделе интеграции аппаратных и программных средств рассматривается планирование и фазы этого процесса, контроль конфигурации и верификации системы, исправление дефектов и ошибок. Выделены аттестация качества ПС и системы, и представление отчетов о достигнутом качестве и функциональной безопасности. Рекомендуется формализовать процедуры сопровождения и модификации для обеспечения их корректности. Для обеспечения функциональной безопасности, обращается внимание на необходимость применения тренажеров и обучения операторов, а также периодических испытаний ПС и системы. В шести Приложениях детализируются процессы и рекомендации по описанию требований к ПС и их реализации, по планированию и обеспечению характеристик качества и функциональной безопасности, по выбору языка программирования, транслятора, редактора связей, а также по тестированию ПС. В подробных таблицах иллюстрируются процессы разработки ПС и их документирования. Содержание этой части стандарта не отличается методичностью изложения рекомендаций и процессов жизненного цикла комплексов программ и ее вряд ли целесообразно использовать при наличии представленных выше, более современных стандартов в области функциональной безопасности.

Основное содержание **второй части** стандарта базируется на требованиях МАГАТЭ по глубоко эшелонированной защите ПС и системы от отказов функционирования при отсутствии предупрежденных негативных воздействий. Отмечается специфика программных дефектов и ошибок, со-

стоящая в единичности и непредсказуемости положения и последствий, что требует их практически полного исключения и отличает от дефектов в аппаратуре, надежность и безопасность которой может рассчитываться аналитически. Рекомендуется создавать систему защитных барьеров и самоконтроля исполнения программ для обеспечения гарантированной работоспособности и безопасности систем управления на базе ПС. Предлагается совокупность методов предотвращения ошибок в ЖЦ ПС путем: создания разнообразия условий функционирования, N-версионного программирования, дублирования спецификаций компонентов при одинаковых функциях, доказательства формальной корректности программ. Специальный раздел посвящен развитию рекомендаций предыдущей части стандарта по выбору программного инструментария, обеспечивающего минимизацию дефектов и ошибок в программах по всем этапам ЖЦ ПС и системы. В отдельный раздел выделены процессы повторного использования ранее разработанных и апробированных на подобных системах компонентов комплексов программ. Рекомендуется проводить детальную оценку функций, качества, результатов опытной эксплуатации и совместимость интерфейсов таких компонентов с вновь разработанной частью ПС, а также возможности их последующей модификации и сопровождения. В четырех Приложениях детализируются некоторые положения этой части стандарта.

Стандарт **МЭК 60880** практически не содержит принципиальных или существенных положений, которые не отражены в совокупности современных международных и национальных стандартов. Исключением является раздел защиты от отказов по общим причинам во второй части стандарта, в котором имеется ряд весьма полезных рекомендаций по обеспечению функциональной безопасности. Приведенные выше в данной главе стандарты полнее и более подробно регламентируют на современном уровне обеспечение функциональной безопасности в ЖЦ сложных комплексов программ и позволяют создавать программный продукт высокого качества и безопасности. Поэтому при создании ПС для атомных электростанций целесообразно применять в основном эти новые стандарты.

## 8. Практические правила управления информационной безопасностью в стандарте ISO 17799

В стандарте **ISO 17799** — Управление информационной безопасностью. Практические правила — основное внимание сосредоточено на организационных и административных задачах обеспечения общей безопасности систем. При этом функциональная безопасность даже не упоминается. Однако некоторые рекомендации могут быть полезными при формировании и организации процессов анализа и обеспечения функциональной безопасности, как дополнения к приведенным выше стандартам.

В стандарте рассмотрена инфраструктура безопасности систем. Рекомендуются совещания руководства по проблемам защиты информации для координации действий разработчиков и распределения обязанностей по обеспечению информационной безопасности. Рекомендации специалистам содержат предложения по независимому анализу безопасности, а также по инвентаризации информационных ресурсов и классификации информации. Выделены анализ и задание требований к безопасности, проверка достоверности входных данных и внутренней обработки данных.

Отмечается необходимость регламентирования обеспечения безопасности в должностных инструкциях и при выделении ресурсов, а также обучение специалистов правилам информационной безопасности, реагированию на события, таящие угрозу безопасности и уведомлению об отказах программного обеспечения. Разделены физическая безопасность системы и безопасность окружающей среды, программных средств разработки и рабочих программ. Значительное внимание уделено администрированию безопасности компьютерных систем и вычислительных сетей, документированию операционных процедур и работе со сторонними организациями. Отмечается необходимость планирования перехода на аварийный режим и

слежения за окружающей средой, за доступом к системам и их использованием. Описаны процедуры управления процессом внесения изменений, технический анализ изменений, вносимых в операционную систему, ограничения на внесение изменений в программы, а также техническая проверка на соответствие иным стандартам безопасности информационных систем и программных средств.

## Литература

1. Трубочев А.П., Долинин М.Ю., Кобзарь М.Т. и др. Оценка безопасности информационных технологий. Общие критерии. Под ред. В.А. Галатенко. М.: СИП РИА, 2001.
2. Устинов Г.Н. Основы информационной безопасности систем и сетей передачи данных. М.: СИНТЕГ. 2000.
3. Липаев В.В. Методы обеспечения качества крупномасштабных программных средств. М.: СИНТЕГ. 2003.
4. Оценка и аттестация зрелости процессов создания и сопровождения программных средств и информационных систем (ISO/IEC TR 15504 — CMM). М.: Книга и бизнес. 2001.
5. Smith D, Simpson K. Functional Safety (A Straightforward Guide IEC 61508 and Related Standards) — Oxford: Planta Tree, 2001.
6. Липаев В.В. Функциональная безопасность программных средств. М.: СИНТЕГ. 2004.

# Jet Info

ИНФОРМАЦИОННЫЙ БЮЛЛЕТЕНЬ

Издается с 1995 года

Издатель: компания Джет Инфо Паблшер

Главный редактор: Дмитриев В.Ю. ([vlad@jet.msk.su](mailto:vlad@jet.msk.su))  
Технический редактор: Овчинникова Г.Ю. ([galya@jet.msk.su](mailto:galya@jet.msk.su))  
Россия, 127015, Москва, Б. Новодмитровская, 14/1  
тел. (095) 411 76 01  
факс (095) 411 76 02  
email: [JetInfo@jet.msk.su](mailto:JetInfo@jet.msk.su) <http://www.jetinfo.ru>

Подписной индекс по каталогу Роспечати

**32555**

