Разработка безопасного программного обеспечения

Эшелон
комплексная безопасность

План выступления

1. Востребованность
2. Анализ известных мер разработки безопасного ПО
3. Определение базового набора требований к разработке безопасного ПО
4. Концептуальная модель выбора мер разработки безопасного ПО

Современные атаки основаны на использовании уязвимостей

Современные атаки

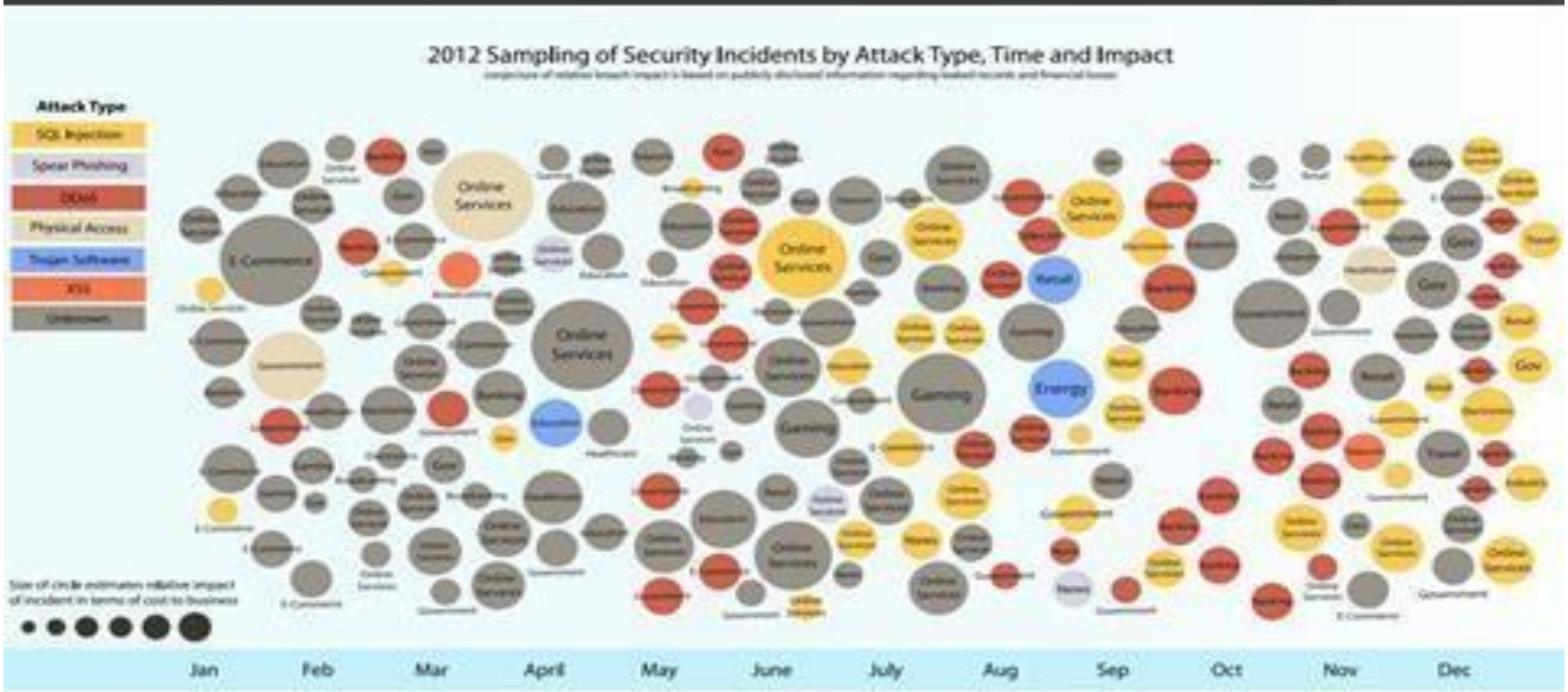
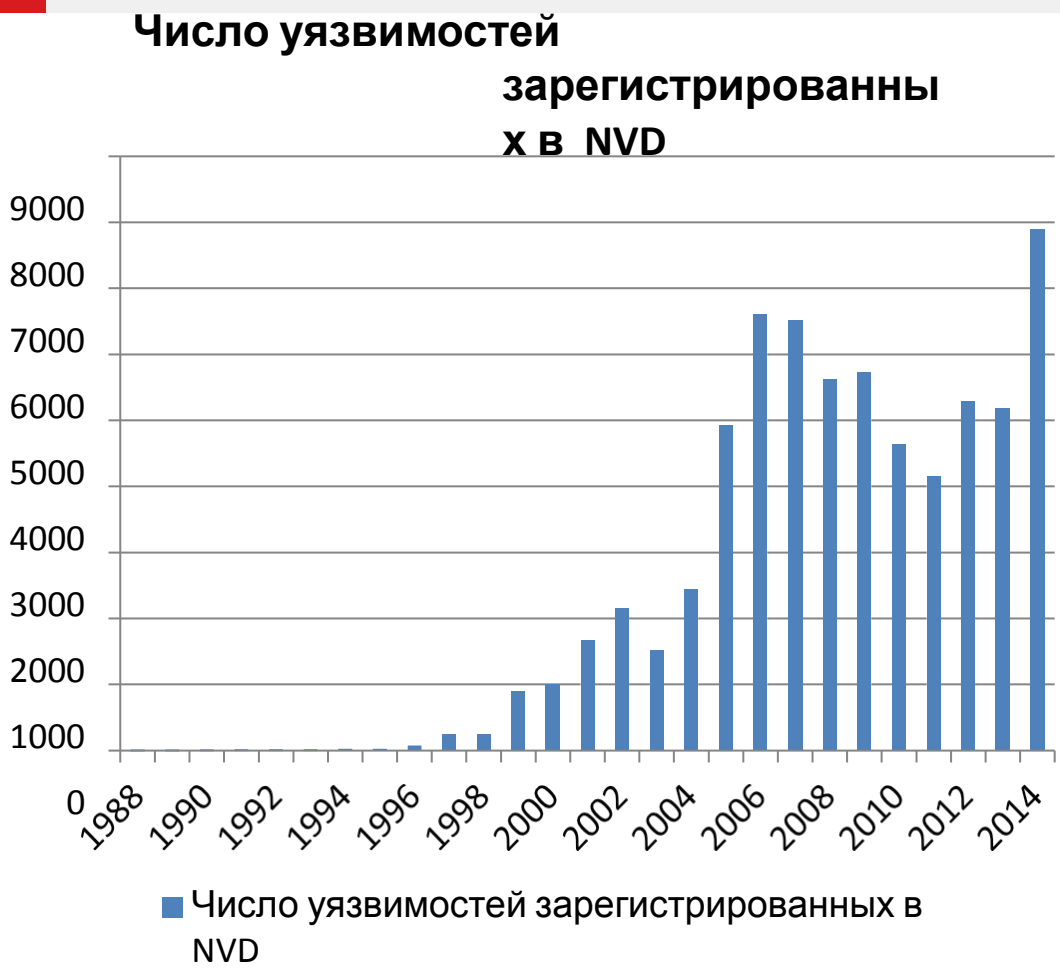


Figure 3: 2012 Sampling of Security Incidents by Attack Type, Time and Impact

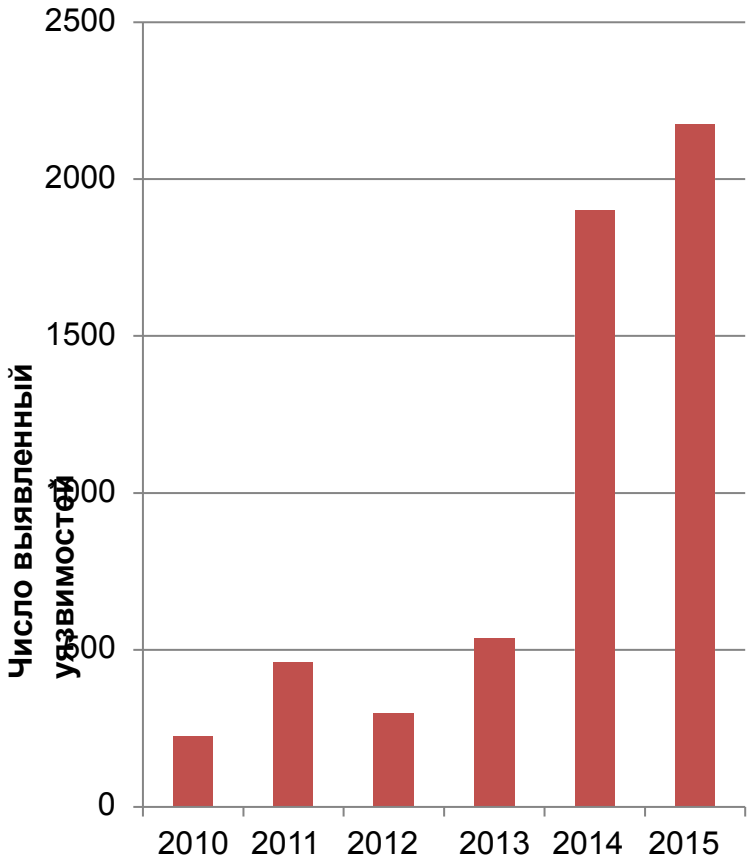
Данные из отчета IBM X-Force 2012 Trend and Risk Report (Март 2013)

Эшелон

Число уязвимостей не уменьшается



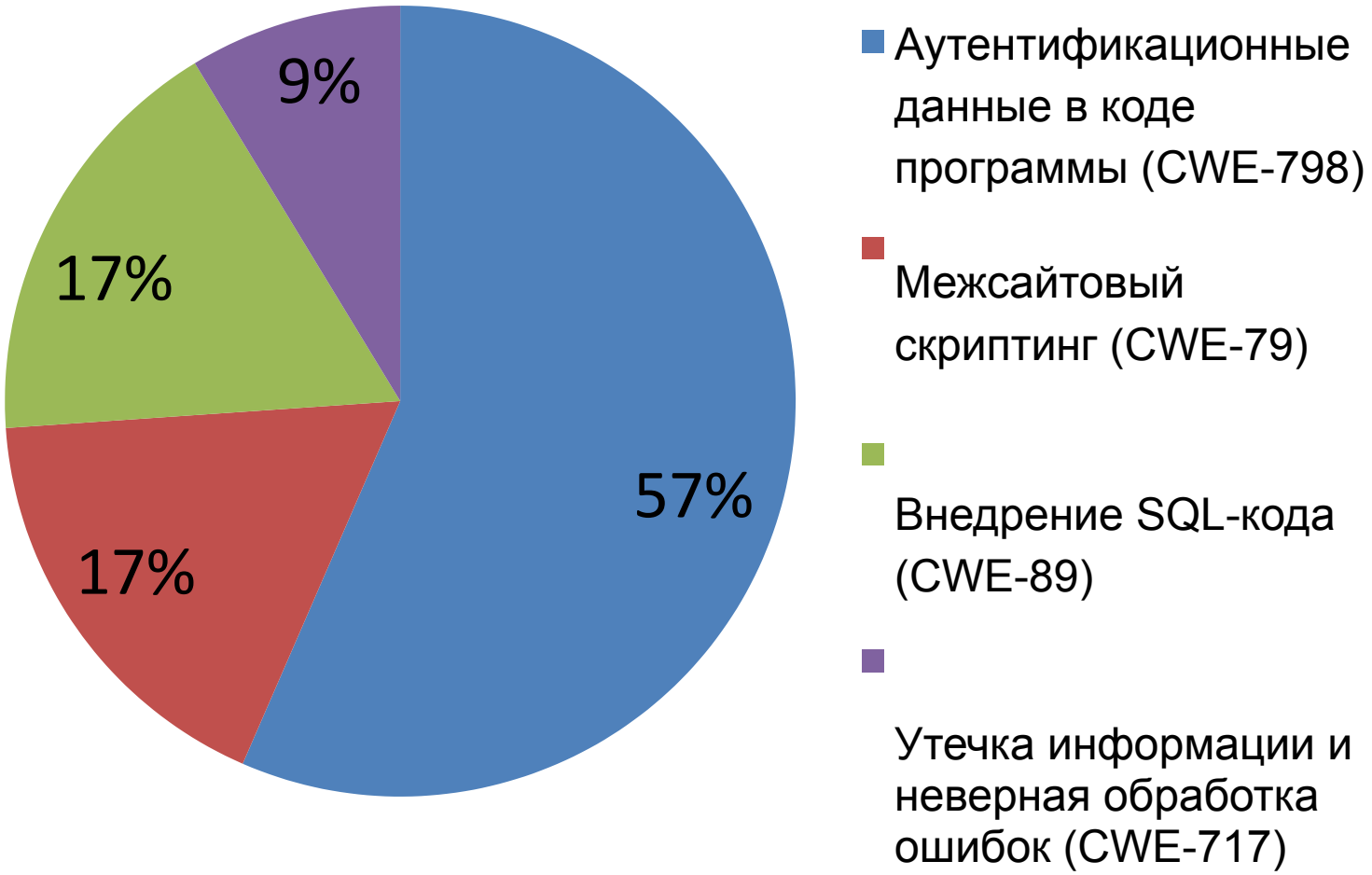
Источник: National Vulnerability Database



Источник: Банк данных угроз безопасности информации ФСТЭК России

Эшелон
комплексная безопасность

Присутствуют как преднамеренные, так и непреднамеренные



Эшелон
комплексная безопасность

Источник: ИЛ НПО «Эшелон»

Нормативный пробел

- Нет регламентации спецэкспертиз при лицензировании
- Нет регламентации проверки производства при оценке соответствия (сертификации)
- Нет рекомендаций разработчикам по разработке безопасного ПО, в т.ч. в рамках СМИБ/СМК

Статистика

- Внедрение СМ БПО позволяет сократить число уязвимостей более чем на 80% (Microsoft SDLC)
- Число ошибок в СЗИ, разработчики которых не имеют международные сертификаты на СМК, в 5 раз больше, чем в СЗИ, разработчики которых имеют международные сертификаты на СМК (НПО «Эшелон)

Цели и задачи исследования

Цель: создание аппарата, позволяющего разработчикам ПО обоснованно формировать множество мер разработки безопасного ПО и проводить с привлечением независимых организаций оценку соответствия применяемых мер требованиям по разработке безопасного ПО

1. Анализ существующих мер, направленных на уменьшение количества уязвимостей в разрабатываемом ПО

2. Формирование базового набора требований к разработке безопасного ПО

3. Разработка концептуальной модели анализа и синтеза мер разработки безопасного ПО

Терминологический аппарат

- Безопасное ПО
- Дефекты (недостатки, слабости, изъяны)

Международная и отечественная практики

Международная практика

- Microsoft SDL
- Cisco SDL
- BSIMM
- ISO/IEC 27034
- Common Criteria
- PA-DSS
- Рекомендации МО США
- ….

Отечественная практика

- РС БР ИББС-2.6-2014
- ГОСТ Р ИСО/МЭК 15408
- ГОСТ Р ИСО/МЭК 27034

Эшелон
комплексная безопасность

Меры по разработке безопасного программного обеспечения (1)

Документы

- *«Общие критерии»*
- *Документы МО США*
- *ISO/IEC TR 24772*
- *ISO/IEC 27034-1*
- *РС БР ИББС-2.6-2014*

Методологии

- *Microsoft SDL*
- *BSIMM*
- *OWASP CLASP*
- *Open SAMM*
- *Cisco SDL*

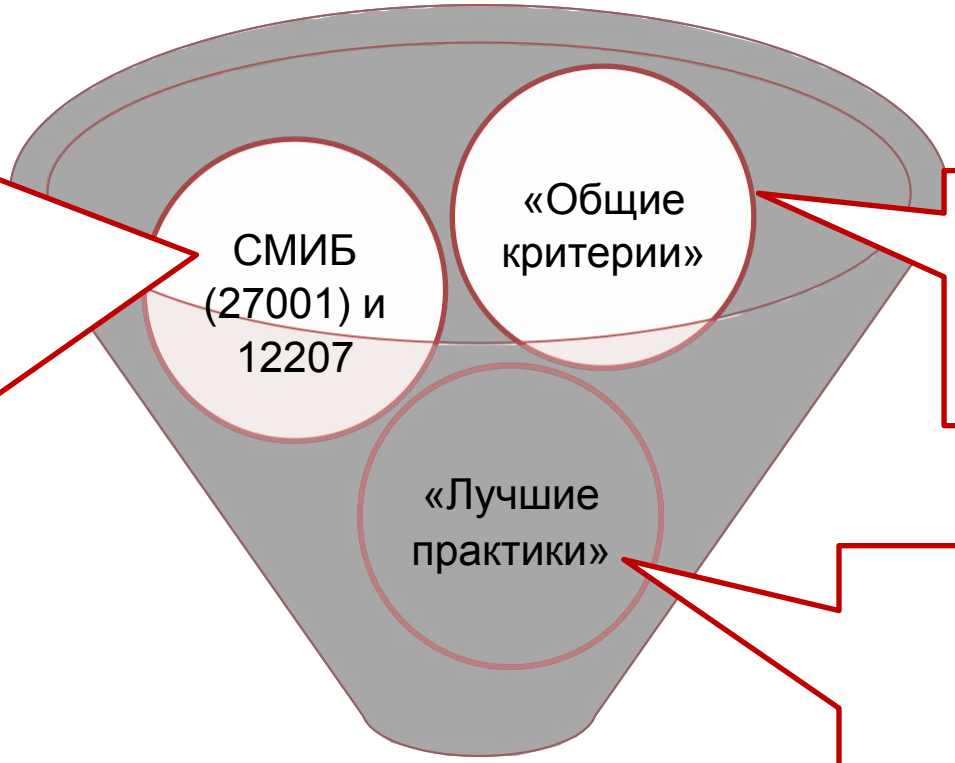
Меры по разработке безопасного программного обеспечения

Результаты анализа известных мер разработки безопасного ПО (2)

1. Не определены требования к действиям аудиторов (оценщиков), проверяющих выполнение требований к разработке безопасного ПО
2. Ограничения стандарта ISO/IEC 15408:
   - стандарт применяется только для ПО с функциями безопасности
   - предлагаемая в стандарте номенклатура мер разработки ПО не учитывает ряд мер (например, обучение специалистов, фаззинг-тестирование, статический анализ)

Учитываемые особенности

Возможность интеграции с СМИБ, согласованность с процессами жизненного цикла по ГОСТ Р ИСО/МЭК 12207



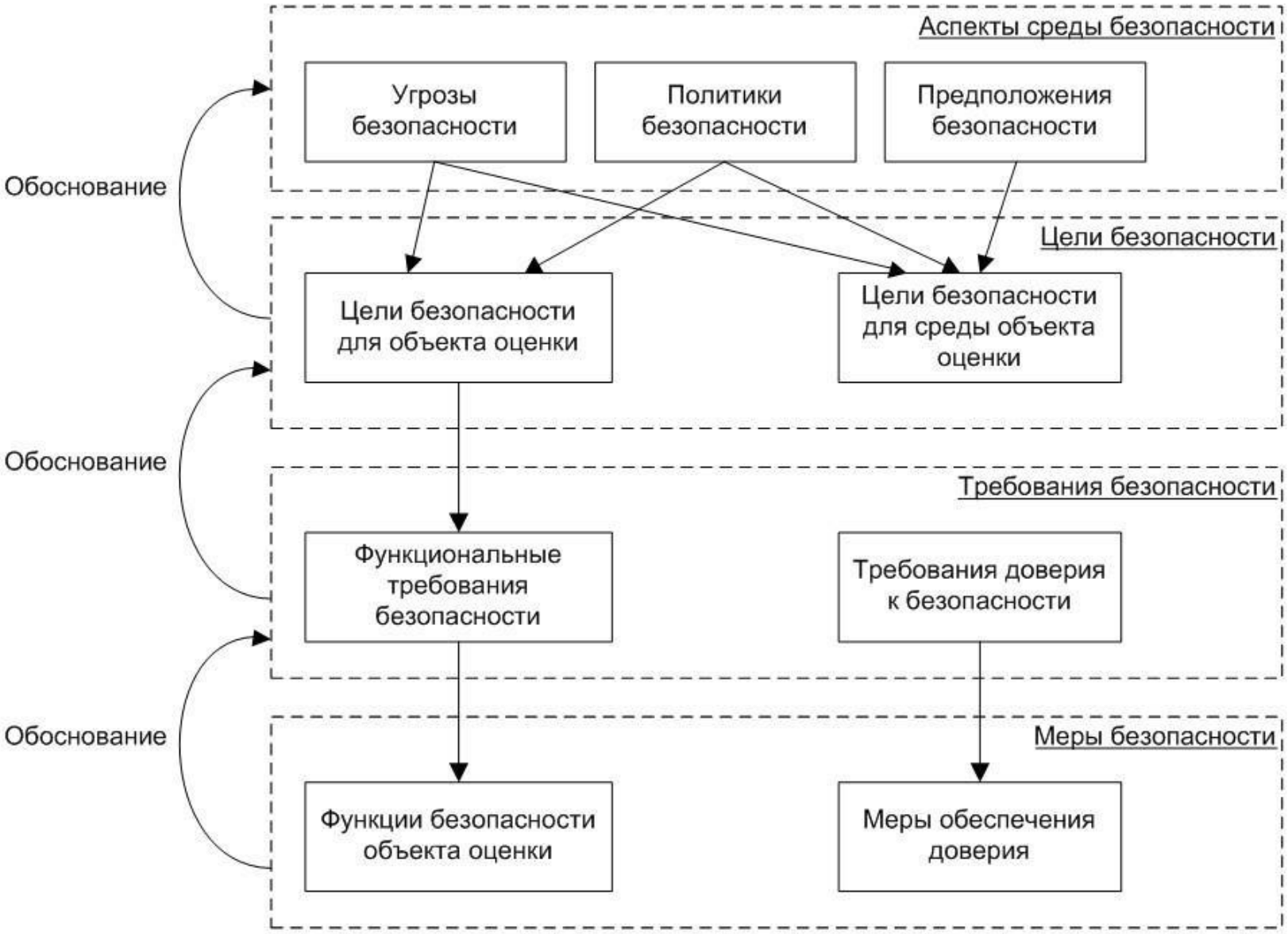
Совместимость с ГОСТ Р ИСО/МЭК 15408

Обеспечение внедрения необходимых процедур на самых ранних стадиях жизненного цикла

Разрабатываемые меры

Эшелон
комплексная безопасность

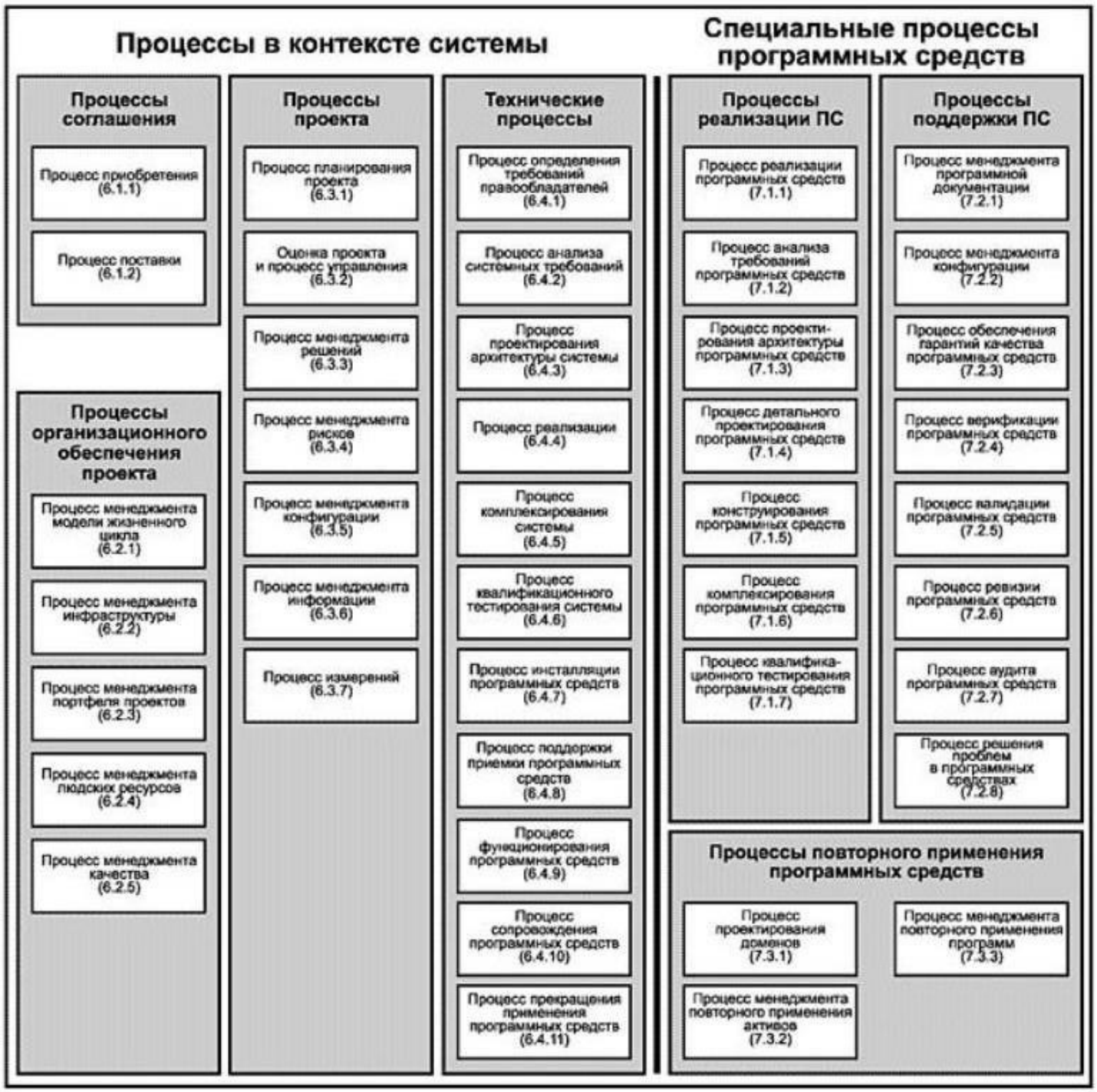
Подход стандарта ИСО/МЭК 15408



Эшелон
комплексная безопасность

Источник: ISO/IEC 15408

Процессы жизненного цикла ПО по ISO/IEC 12207



Эшелон
комплексная безопасность

Группы процессы ПО по ISO/IEC 12207

7	*Процессы жизненного цикла программных средств*
7.1	*Процессы реализации программных средств*
7.1.1	*Процесс реализации программных средств*
7.1.2	*Процесс анализа требований к программным средствам*
7.1.3	*Процесс проектирования архитектуры программных средств*
7.1.4	*Процесс детального проектирования программных средств*
7.1.5	*Процесс конструирования программных средств*
7.1.6	*Процесс комплексирования программных средств*
7.1.7	*Процесс квалификационного тестирования программных средств*
7.2	*Процессы поддержки программных средств*
7.2.1	*Процесс менеджмента документации программных средств*
7.2.2	*Процесс менеджмента конфигурации программных средств*
7.2.3	*Процесс обеспечения гарантии качества программных средств*
7.2.4	*Процесс верификации программных средств*
7.2.5	*Процесс валидации программных средств*
7.2.6	*Процесс ревизии программных средств*
7.2.7	*Процесс аудита программных средств*
7.2.8	*Процесс решения проблем в программных средствах*
7.3	*Процессы повторного применения программных средств*
7.3.1	*Процесс проектирования доменов*
7.3.2	*Процесс менеджмента повторного применения активов*
7.3.3	*Процесс менеджмента повторного применения программ*

Гармонизация с ГОСТ Р ИСО 15408

Меры по разработке безопасного ПО	Семейство (класс) требований доверия к безопасности по ГОСТ Р ИСО/МЭК 15408-3
5.1.3.1	ASE «Оценка задания по безопасности»
5.2.3.1	ADV_ARC «Архитектура безопасности»
5.2.3.2	ADV_FSP «Функциональная спецификация», ADV_TDS «Проект ОО», ADV_ARC «Архитектура безопасности»
5.3.3.1	ALC_TAT «Инструментальные средства и методы»
5.3.3.2	ADV_IMP «Представление реализации»
5.3.3.3	ALC_TAT «Инструментальные средства и методы» (в части ALC_TAT.2)
5.3.3.4	ALC_TAT «Инструментальные средства и методы» ALC_CMC «Возможности управления конфигурацией» (в части ALC_CMC.5)
5.3.3.5	ALC_TAT «Инструментальные средства и методы» (в части ALC_TAT.2), ALC_CMC «Возможности управления конфигурацией» (в части ALC_CMC.5), ALC_CMS «Область управления конфигурацией» (в части ALC_CMS.4)
5.4.3.1	ATE_COV «Покрытие», ATE_DPT «Глубина», ATE_FUN «Функциональное тестирование»
5.4.3.2	AVA_VAN «Анализ уязвимостей»
5.4.3.3	отсутствует
5.5.3.1	ALC_DEL «Поставка»
5.5.3.2	AGD_OPE «Руководство пользователя по эксплуатации», AGD_PRE «Подготовительные процедуры»
5.6.3.1	ALC_FLR «Устранение недостатков»
5.6.3.2	ALC_FLR «Устранение недостатков» (в части ALC_FLR.2)
5.6.3.3	ALC_FLR «Устранение недостатков» (в части ALC_FLR.2)
5.6.3.4	отсутствует
5.6.3.5	ALC_TAT «Инструментальные средства и методы», ALC_CMC «Возможности управления конфигурацией, ALC_CMS «Область управления конфигурацией»
5.7.3.1	ALC_CMC «Возможности управления конфигурацией», ALC_CMS «Область управления конфигурацией»
5.7.3.2	ALC_CMC «Возможности управления конфигурацией», ALC_CMS «Область управления конфигурацией»
5.7.3.3	ALC_CMC «Возможности управления конфигурацией»
5.7.3.4	ALC_CMC «Возможности управления конфигурацией»
5.8.3.1	ALC_CMC «Возможности управления конфигурацией»
5.8.3.2	ALC_DVS «Безопасность разработки»
5.8.3.3	ALC_CMC «Возможности управления конфигурацией»
5.9.3.1	отсутствует
5.9.3.2	отсутствует

Эшелон
комплексная безопасность

Результаты разработки базового набора требований: номенклатура (1)

Процесс ЖЦ по ИСО/МЭК 12207	Предлагаемые требования
Процесс анализа требований к ПО	Определение требований к ПО
Процесс проектирования и детального проектирования архитектуры ПО	Моделирование угроз БИ
	Разработка архитектуры ПО с учетом результатов моделирования угроз
Процессы конструирования и комплексирования ПО	Идентификация инструментальных средств разработки ПО
	Использование стандарта оформления исходного кода программы
	Статический анализ исходного кода программы
	Периодическая экспертиза исходного кода программы

Результаты разработки базового набора требований: номенклатура (2)

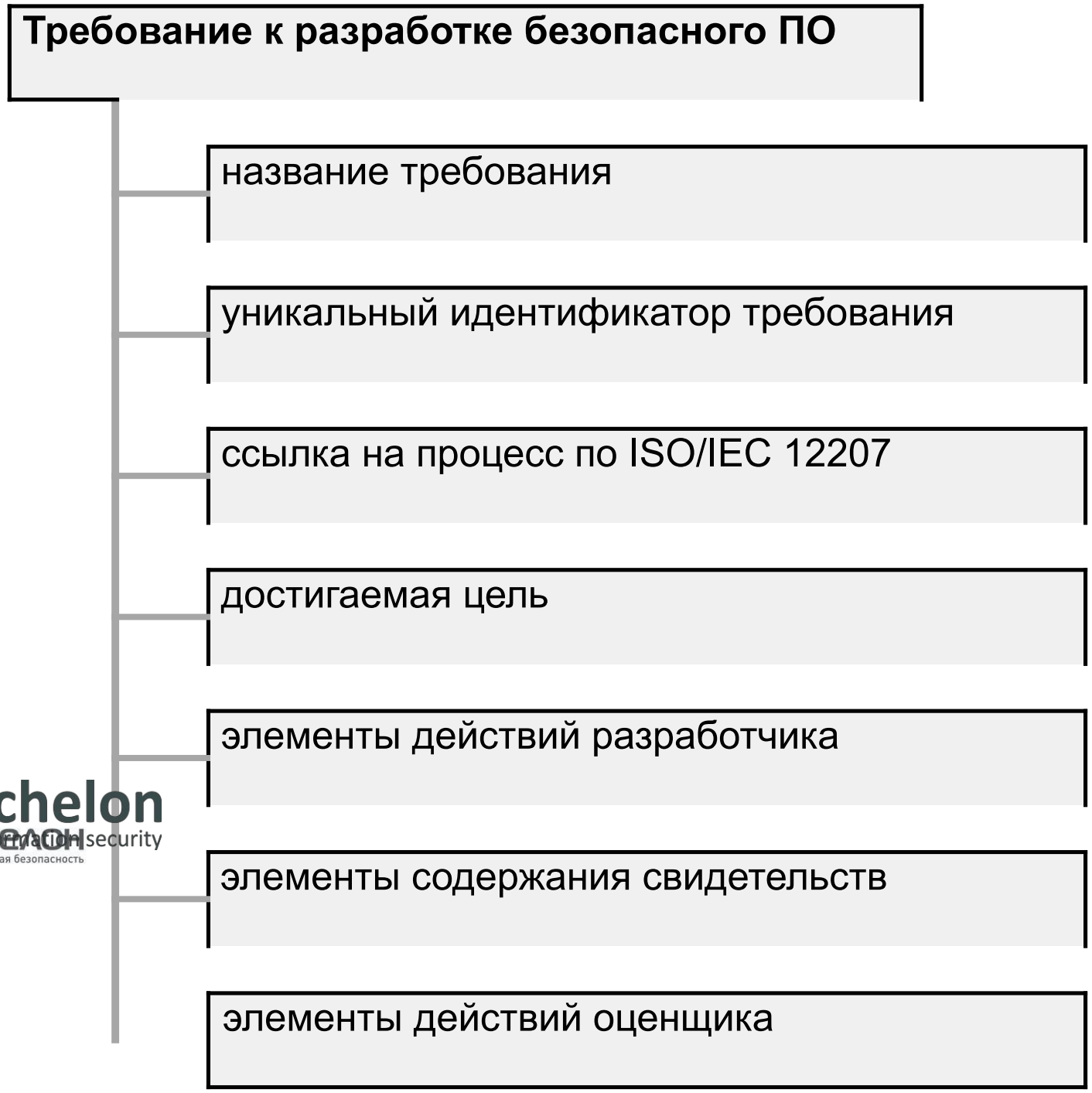
Процесс ЖЦ по ИСО/МЭК 12207	Предлагаемые требования
Процесс квалификационного тестирования ПО	Функциональное тестирование ПО
	Тестирование на проникновение
	Динамический анализ кода программы
Процесс решения проблем в ПО	Отслеживание и исправление обнаруженных уязвимостей ПО и ошибок ПО
Процесс менеджмента документации и конфигурации ПО	Использование системы управления конфигурациями

Echelon
information security
комплексная безопасность

Результаты разработки базового набора требований: номенклатура (3)

Процесс ЖЦ по ИСО/МЭК 12207	Предлагаемые требования
Процесс менеджмента инфраструктуры	Защита от несанкционированного доступа к элементам конфигурации
	Резервное копирование и восстановление элементов конфигурации
	Регистрацию событий
Процесс менеджмента людских ресурсов	Обучение сотрудников

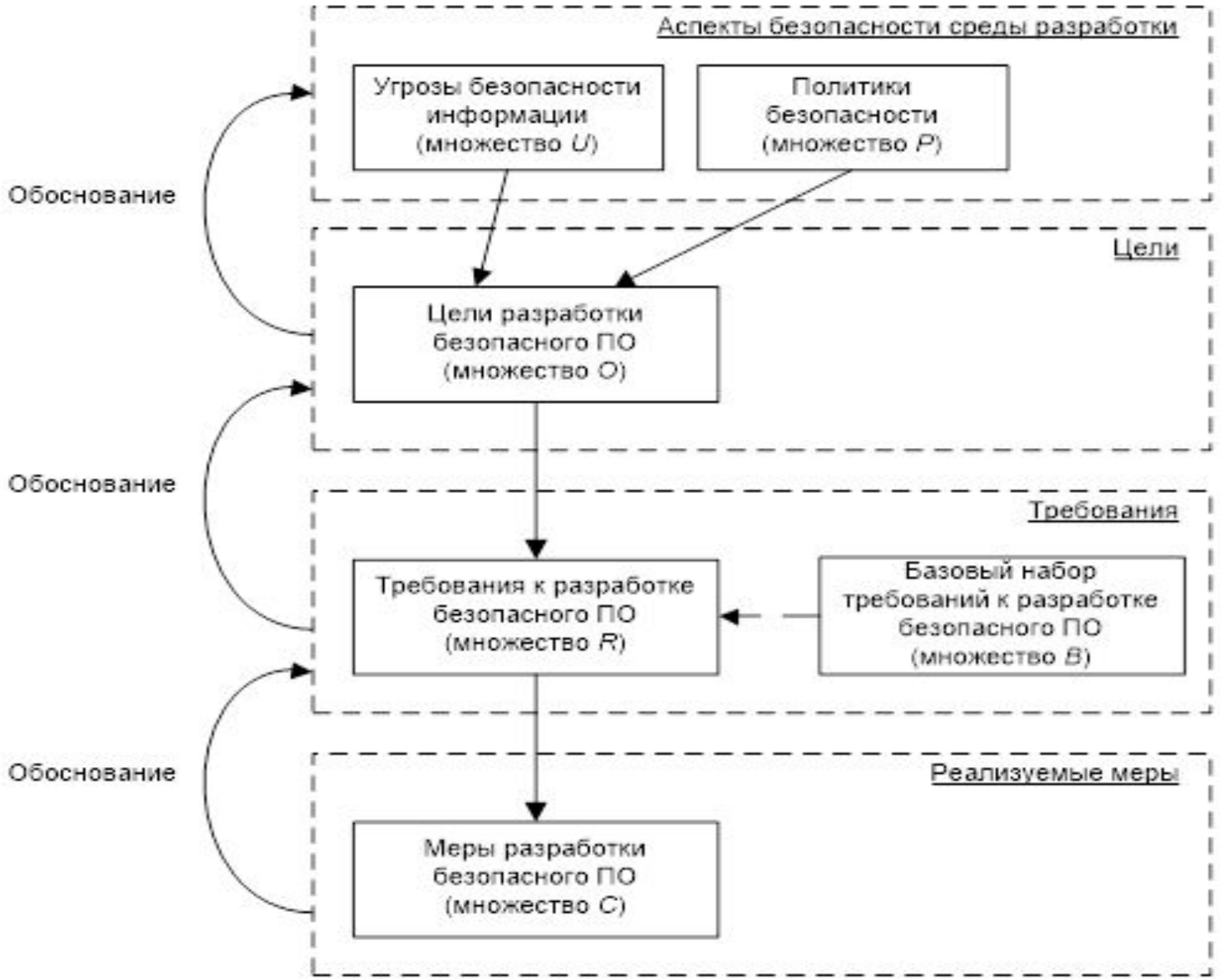
Предлагаемый формат представления



Пример

Параметр	Значение параметра
Название	Статический анализ исходного кода программы
Идентификатор требования	КК-4
Процесс жизненного цикла ПО	Процессы конструирования и комплексирования ПО
Цель	- выявление и устранение потенциально уязвимых конструкций в исходном коде программы; - формирование исходных данных для выполнения задач динамического анализа и тестирования на проникновение в рамках процесса квалификационного тестирования ПО.
Элементы действий разработчика	Разработчик ПО должен проводить статический анализ исходного кода программы с целью выявления потенциально уязвимых конструкций в исходном коде программы. Статический анализ исходного кода программы следует проводить в отношении заимствованных у сторонних разработчиков ПО компонентов, если для них доступен исходный код программы. По результатам статического анализа исходного кода программы может проводиться доработка программы. При отсутствии необходимости в такой доработке или невозможности доработки программы разработчик должен документировать обоснование этого факта.
Элементы содержания и представления документированных свидетельств	Документированные свидетельства статического анализа исходного кода программы должны содержать: - сведения о периодичности проведения статического анализа исходного кода программы; - наименование и идентификационные признаки инструментальных средств, используемых для проведения статического анализа исходного кода программы; - список выявленных потенциально уязвимых конструкций в исходном коде программы (при выявлении), описание действий, направленных на их устранение, или обоснование невозможности или отсутствия необходимости в доработке программы.
Элементы действий оценщика	1. Оценщик должен исследовать представленные свидетельства и подтвердить, что они удовлетворяют предъявляемым требованиям. 2. Оценщик должен сделать независимое заключение, что разработчик выполняет статический анализа исходного кода программы по результатам опроса работников организации-разработчика ПО, имеющих отношение к разработке ПО, анализа среды разработки ПО.
Примечание	Статический анализ исходного кода программы выполняется разработчиком ПО или сторонними организациями, обладающими компетенцией в области выявления уязвимостей ПО, для актуальной версии кода программы. Статический анализ исходного кода программы позволяет выполнить поиск потенциально уязвимых конструкций в исходном коде программы, которые могут привести к наличию уязвимости ПО, а также проверять соответствие исходного кода программы принятому в организации стандарту оформления исходного кода программы.

Подход стандарта ИСО/МЭК 15408



Эшелон
комплексная безопасность

Источник: НПО Эшелон

Разработанная концептуальная модель выбора мер

Модель характеризуется кортежем $\langle B, U, P, O, R, C, F_O, F_R, F_C \rangle$:

- множество B – базовый набор мер разработки безопасного ПО;
- множество U – множество угроз безопасности информации, которые являются актуальными для среды разработки ПО;
- множество P – множество идентифицированных положений политик безопасности, которым должна соответствовать среда разработки ПО;
- множество O – множество целей разработки безопасного ПО
- множество R – множество требований разработки безопасного ПО
- множество C – множество мер разработки безопасного ПО
- $F_O: U \cup P \to O$ - процедура формирования целей разработки безопасного ПО;
- $F_R: B \cup O \to R$ – процедура выбора требований по разработке безопасного ПО;
- $F_C: R \to C$ - процедура синтеза мер разработки безопасного ПО.

Разработанная методика выбора мер

Этап 1. Идентификация и описание аспектов безопасности среды разработки ПО

Шаг 1. Формирование множества $U = \{u_1, u_2, \dots u_a\}$ угроз безопасности информации, которые являются актуальными для среды разработки ПО.

Шаг 2. Формирование множества $P = \{p_1, p_2, \dots, p_b\}$ положений политик безопасности, которым должна соответствовать среда разработки ПО.

Этап 2. Формирование и обоснование множества целей разработки безопасного ПО

Шаг 3. Формирование множества целей разработки безопасного ПО: $O = F_O(U, P)$.

Шаг 4. Выполнение обоснования полноты и достаточности сформулированного множества целей

Этап 3. Выбор и обоснование требований по разработке безопасного ПО

Шаг 5. Формирование множества требований по разработке безопасного ПО на основе базового набора требований с учетом необходимости достижения идентифицированных целей: $R = F_R(B, O)$.

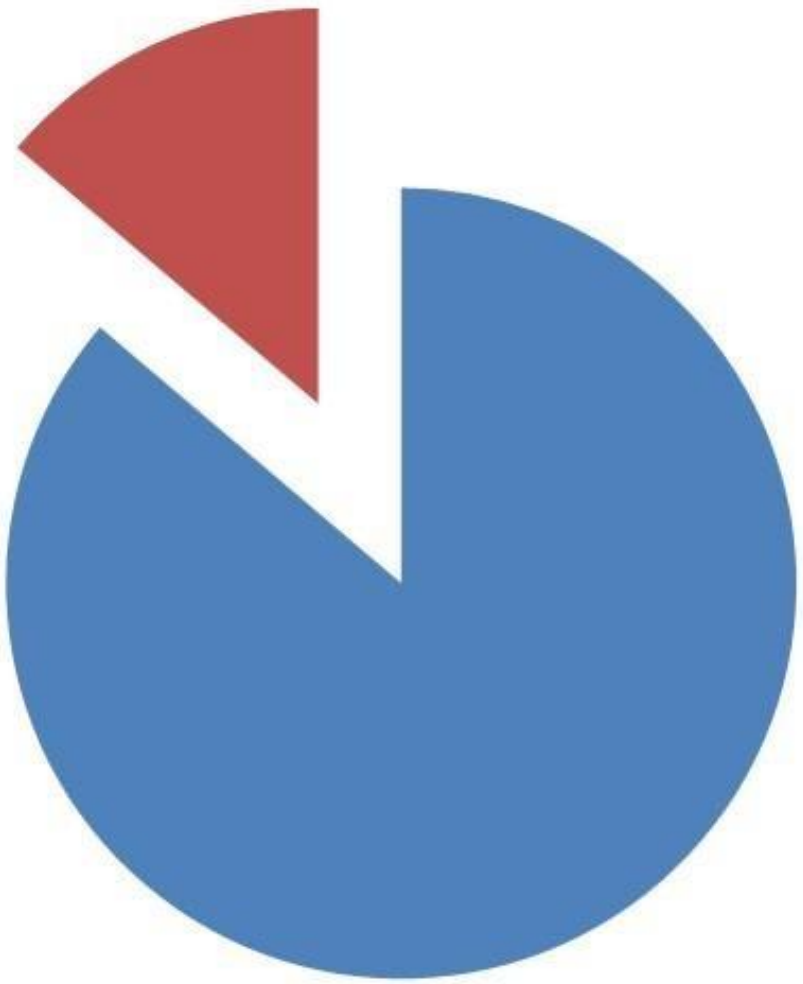
Шаг 6. Выполнение обоснования полноты и достаточности выбранного множества требований

Этап 4. Синтез и обоснование мер разработки безопасного ПО

Шаг 7. Формирование множества мер разработки безопасного ПО, планируемых применять в среде разработки ПО: $C = F_C(R)$.

Шаг 8. Выполнение обоснования полноты и достаточности синтезированного множества мер разработки

Оценка эффективности: уязвимости ПО, выявляемые при проведении сертификации

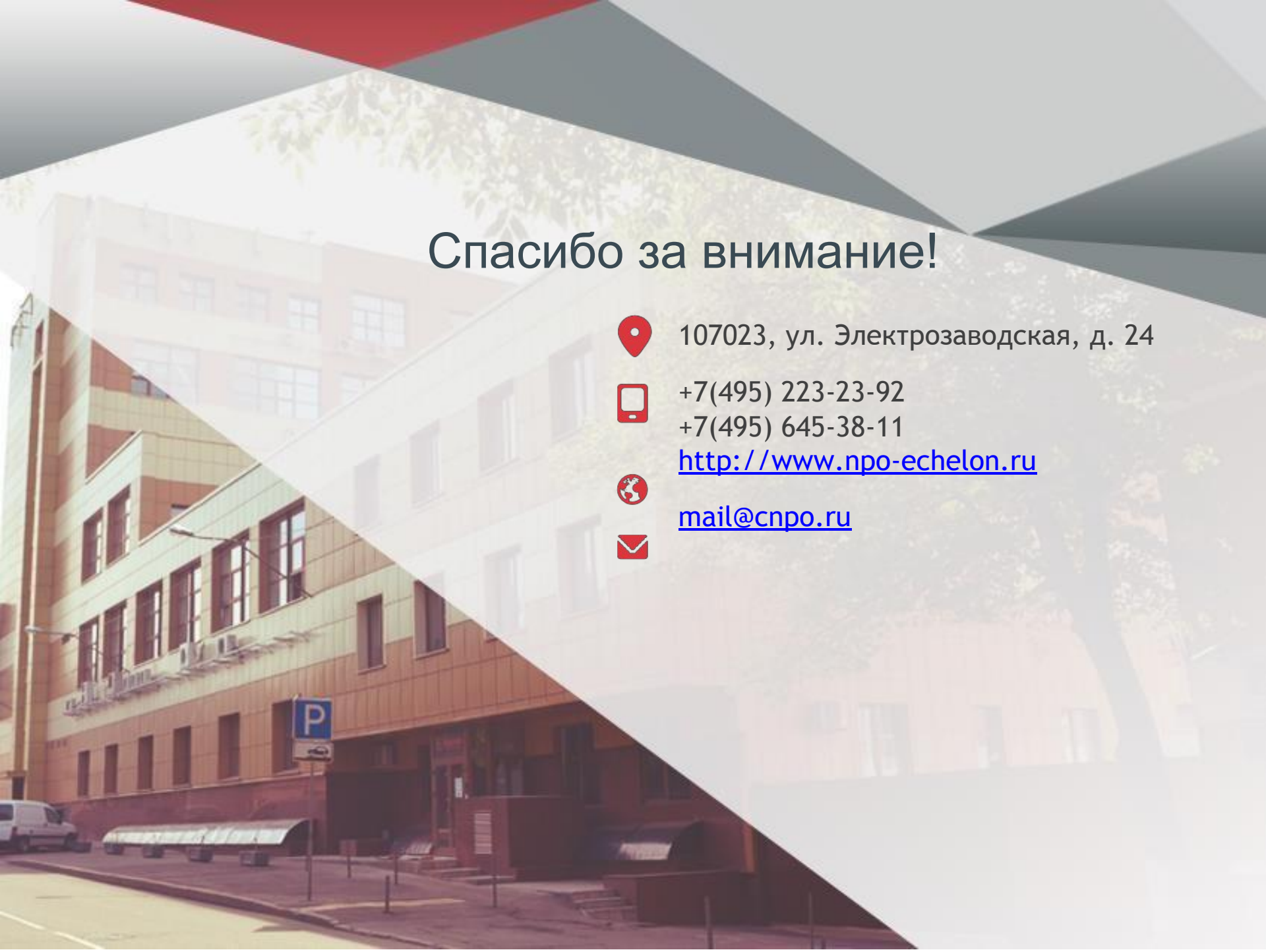


Источник: ИЛ НПО «Эшелон»

Заключение

1. Разработан базовый набор из 24-х требований по разработке безопасного ПО.
2. Предложены концептуальная модель и методика выбора мер разработки безопасного ПО, которая обеспечивает возможность обоснованного выбора мер разработки безопасного ПО и обладает свойством согласованности со стандартами серии «Общие критерии»
3. Данный подход положен в основу проекта ГОСТ «Защита информации. Разработка безопасного программного обеспечения. Общие требования».

Спасибо за внимание!



107023, ул. Электрозаводская, д. 24

+7(495) 223-23-92
+7(495) 645-38-11

http://www.npo-echelon.ru

mail@cnpo.ru