IV Конференция «Актуальные вопросы защиты информации»

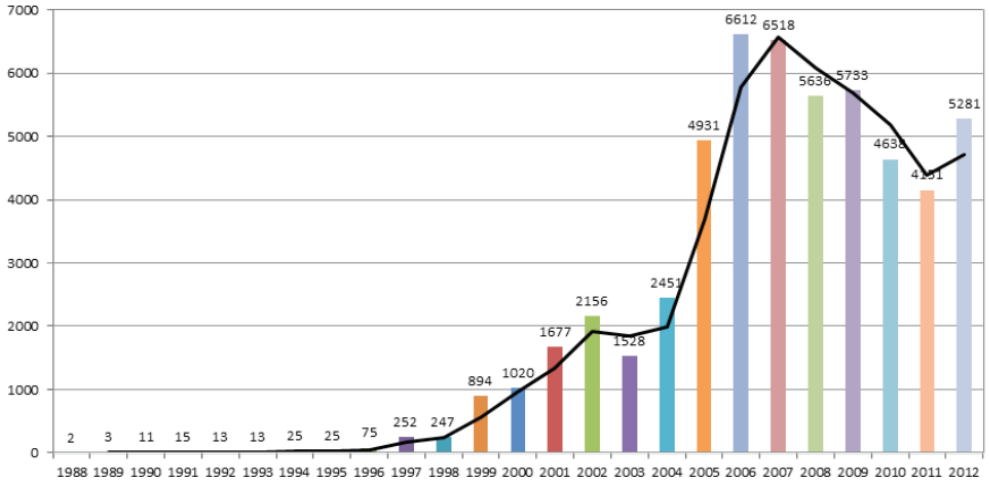
Практический опыт компании ИнфоТеКС в безопасной разработке сертифицированных средств защиты информации. Достижения и пути развития.

Дмитрий Гусев

зам. генерального директора ОАО «ИнфоТеКС»

gusev@infotecs.ru

Информационная система и уязвимости



Статистика выявленных уязвимостей с 1988 по 2012 год в различных программных продуктах (включая СЗИ), используемых в ИС (ист.:SourceFire)

Вывод исследователей – основным типом уязвимостей является эксплуатация ошибок с «переполнением буфера».

Такой тип ошибок в равной степени свойственен всем программным продуктам, включая СЗИ, «закладывается» на этапе разработки и не может быть исправлен без внесения изменений в программный код продукта.

Элементы ИС с установленными СЗИ/СКЗИ

Безопасность ИС:

- Безопасность операционной системы (среды функционирования)
- Безопасность прикладных сервисов
- Безопасность встраиваемых средств защиты информации

infotecs®

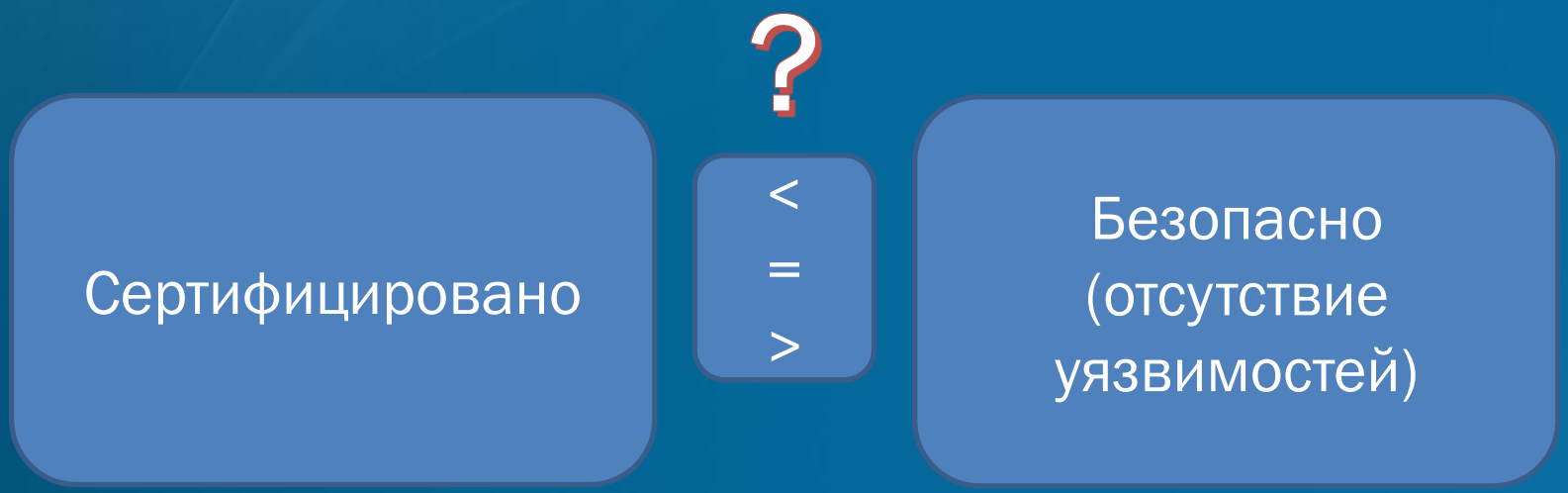
Потребности в СЗИ очевидны, но помимо реализуемых функциональных и нефункциональных характеристик, они влияют на защищаемую систему (изменяют её) наравне с любыми другими сервисами системы.

Требуется оценка взаимного влияния по критериям безопасности

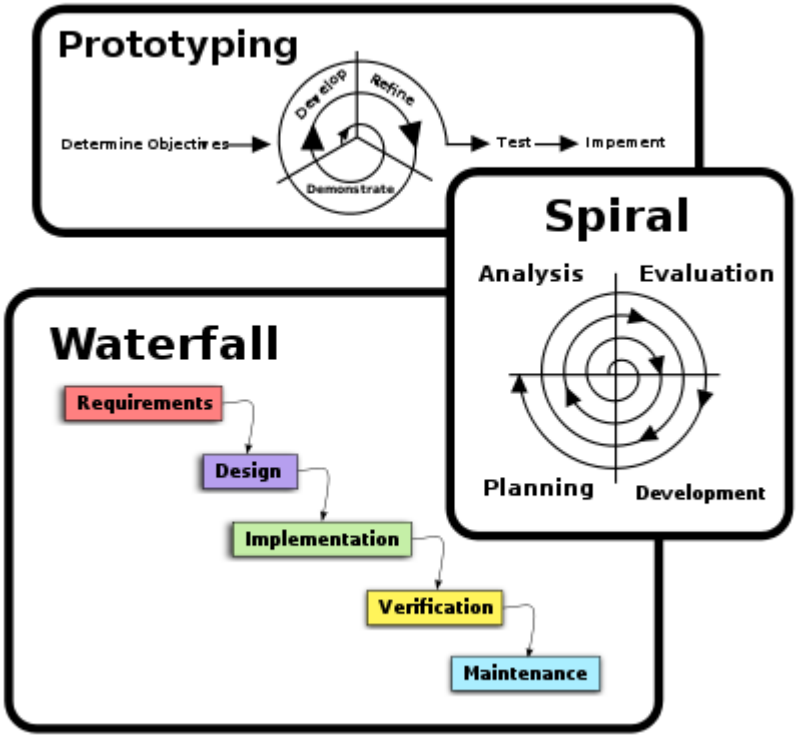
А тем временем…

- Активизация интереса независимых экспертов по ИБ к отечественным СЗИ/СКЗИ
- Расширение перечня публичных мероприятий в России по практической безопасности (ZeroNights, Positive Hack Days и др.)
- Шпионские скандалы в мире информационных технологий (разоблачения от WikiLeaks, Сноудена, …) с раскрытием деталей работы спецслужб по ведению кибервойн

Но как это касается вопросов разработки и эксплуатации сертифицированных СКЗИ/СЗИ?



Как делают ИТ-продукты



Классические типы циклов разработки:

- «Водопад»: линейный процесс
- «Спираль»: линейно-итеративный процесс
- «Приращениями»: линейно-итеративный процесс или V-модель
- «Прототипирование»: итеративный процесс
- «Быстрая разработка» (RAD): итеративный процесс

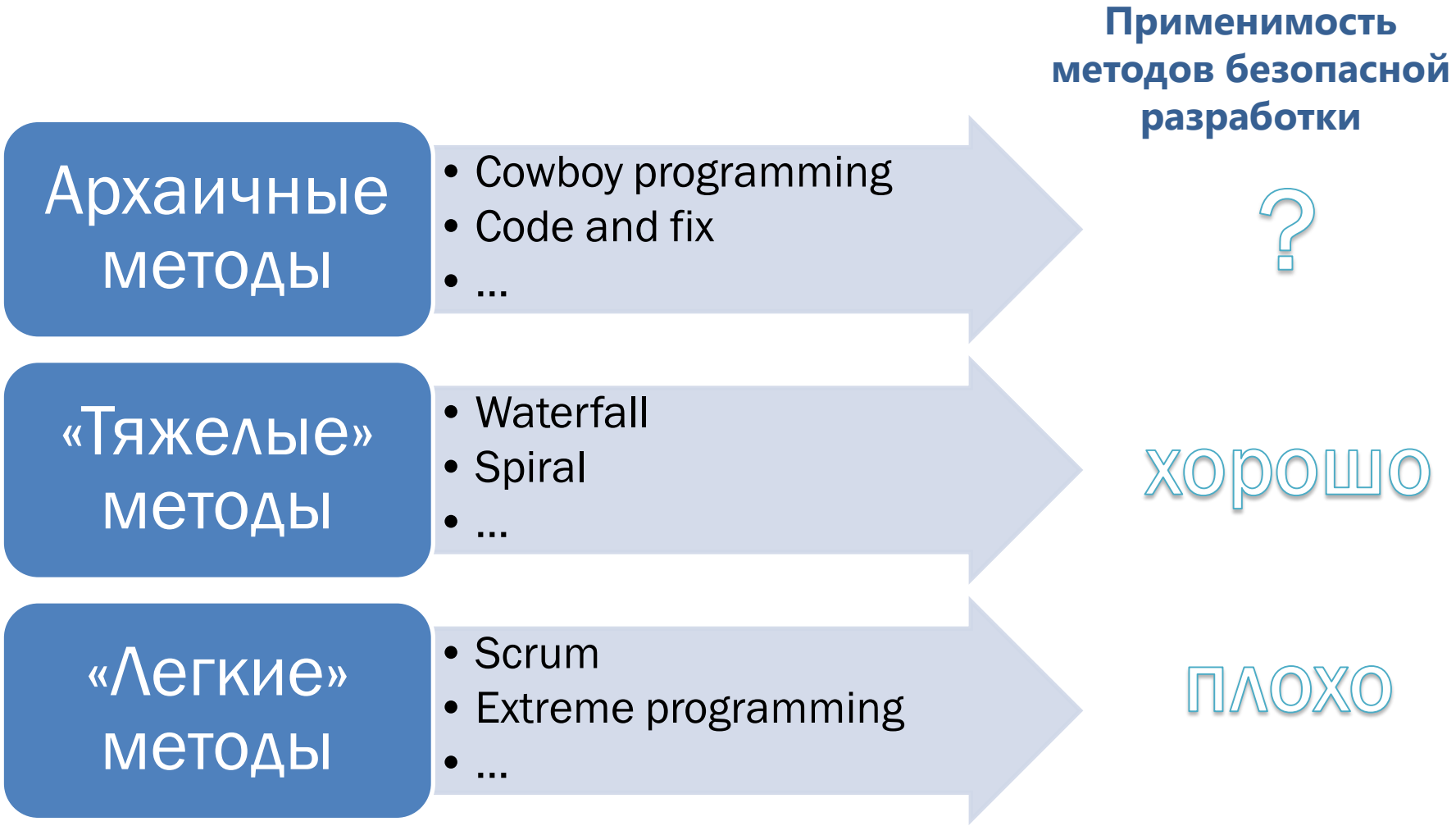
Гибкие (Agile) методологии разработки:

- Scrum
- Экстремальное программирование
- Адаптивная разработка (ASD)
- Динамический метод разработки (DSDM)
- …

Разработка безопасных ИТ-продуктов (Security Development Lifecycle)

Microsoft

Эволюция в методах разработки и SDL



infotecs®

А что по этому поводу установлено регуляторами и предписано нормативными актами?

Документы ФСТЭК России

Приказ ФСТЭК России от 11 февраля 2013 г. N 17 "Об утверждении требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах"

- п. 16. - анализ уязвимостей информационной системы и принятие мер защиты информации по их устранению;
- п. 16.6. Анализ уязвимостей информационной системы проводится в целях оценки возможности преодоления нарушителем системы защиты информации информационной системы и предотвращения реализации угроз безопасности информации.
- Анализ уязвимостей информационной системы включает анализ уязвимостей средств защиты информации и технических средств и программного обеспечения информационной системы.

- п. 18.1. В ходе управления (администрирования) системой защиты информации информационной системы осуществляются:
- - установка обновлений программного обеспечения, включая программное обеспечение средств защиты информации, выпускаемых разработчиками (производителями) средств защиты информации или по их поручению;

Документы ФСТЭК (Гостехкомиссии) России

Руководящий документ. Защита от НСД к информации. Часть 1. Программное обеспечение средств защиты информации. Классификация по уровню контроля отсутствия недекларированных возможностей

Приказ председателя Гостехкомиссии России от 4 июня 1999 года № 114

- **2.1. Недекларированные возможности** - функциональные возможности ПО, не описанные или не соответствующие описанным в документации, при использовании которых возможно нарушение конфиденциальности, доступности или целостности обрабатываемой информации. Реализацией недекларированных возможностей, в частности, являются программные закладки.

- 2.2. **Программные закладки** – преднамеренно внесенные в ПО функциональные объекты, которые при определенных условиях (входных данных) инициируют выполнение не описанных в документации функций ПО, приводящих к нарушению конфиденциальности, доступности или целостности обрабатываемой информации.

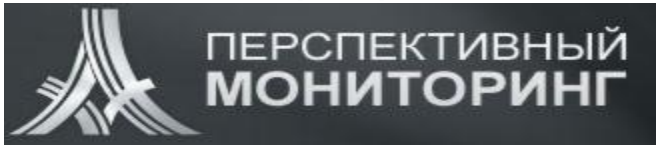
- **Только со 2 уровня контроля отсутствия НДВ установлено требование:**

- синтаксический контроль наличия заданных конструкций в исходных текстах ПО из списка (базы) **потенциально опасных программных конструкций**;

И что по этому поводу делает ИнфоТеКС?

Работы по обеспечению безопасности продуктов ViPNet

- Регулярный внутренний аудит безопасности продуктов ViPNet – поиск и устранение уязвимостей независимой от разработчиков командой компетентных экспертов



- Следование мировой практике взаимодействия с независимыми экспертами по ИБ - публикация политики ответственного разглашения
- Регламент устранения выявленных уязвимостей
- Разработка во взаимодействии с регуляторами и испытательными лабораториями порядка обновлений сертифицированных версий СЗИ/СКЗИ
- Сертификация по ISO9000 СМК по разработке и производству продуктов

Следование подходу SDL при разработке СЗИ ViPNet

	Basic	Standardized (Activities are expert led)	Advanced (Activities are led by central security team)	Dynamic (Activities are co-led by product teams and central security team)
Training, Policy, and Organizational Capabilities	• Setting baseline and goals for SDL	• Executive support: Tacit • Enterprise coverage: Few SDL pilot projects • Training: Basic concepts • Basic security bug tracking	• Executive support: Explicit • Enterprise coverage: New, high-risk projects • Training: Common baseline • Central security team exists	• Executive support: SDL mandated and enforced • Enterprise coverage: All projects with meaningful risk • Training: Custom training • SDL is adapted to the development methodology
Requirements and Design	• Undefined or inconsistent	• Risk assessment • Threat models: Piloted for high-risk modules	• Functional requirements for security and privacy • Standard security solutions • Threat models: Created with expert assistance	• Threat models: Independently created by product groups
Implementation	• Undefined or inconsistent	• Compiler defenses • Banned functions • Cross-site scripting and SQL injection defenses	• Static analysis tools	• In-house, product-specific security tools development and customization
Verification	• Undefined or inconsistent	• File fuzzing • Basic Web application scanning • Penetration testing by third parties as appropriate	• Comprehensive fuzzing and Web application scanning • Threat model validated	• In-house development and customization of tools to: • Detect vulnerabilities • Audit compliance with SDL
Release and Response	• Undefined or inconsistent	• Final Security Review: Verify internal and external compliance • Project archiving • Response: Basic	• Response plan in place, incident tracking • Basic root-cause analysis	• Real-time incident tracking • Advanced root-cause analysis and formal feedback into policy

- Все разработчики СЗИ, придерживающиеся классических методов разработки, начинают с уровня **Basic**
- ИнфоТеКС на 50% уже сейчас выполняет требования уровня **Standardized**
- После создания выделенной команды экспертов в ЗАО «Перспективный мониторинг» ИнфоТеКС начал использовать методы уровня **Advanced**

infotecs®

Спасибо за внимание!
Вопросы?

Дмитрий Гусев
заместитель генерального директора ОАО «ИнфоТеКС»
gusev@infotecs.ru