The background of the slide is a photograph of a businessman in a dark suit and tie, holding a large, metallic, 3D-rendered gear. The gear is semi-transparent, revealing a complex internal mechanism of smaller gears and shafts. The scene is set in a modern office environment with blurred computer monitors and desks in the background.

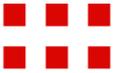
Автоматическая генерация исходных
текстов для повышения качества
программных продуктов

Минко В.С.

Введение

infotecs®

infotecs


VIPNet
Virtual Private Network



Классификация проблем с качеством ПО

С точки зрения информационной безопасности при разработке ПО можно выделить следующие категории проблем:

- Намеренные закладки, внесённые внутренним злоумышленником
- Ненамеренные (случайные) ошибки в ПО
- Ошибки в самих спецификациях



Классификация проблем с качеством ПО

С точки зрения информационной безопасности при разработке ПО можно выделить следующие категории проблем:

- Намеренные закладки, внесённые внутренним злоумышленником
- **Ненамеренные (случайные) ошибки в ПО**
- Ошибки в самих спецификациях



Технологии для исключения ненамеренных ошибок в ПО



Рост «интеллектуальности» компиляторов



```
if (x = 5)
{
    /* some code ... */
}
```

```
somefunc (void)
{
    /* some code ... */
    return;
}
```

Рост «интеллектуальности» компиляторов

test.c: ошибка: по умолчанию возвращаемый тип функции - «int»

```
somefunc(void)
```

^

test.c: ошибка: оператор «return» без значения в функции, возвращающей не void-
значение

```
return;
```

^

test.c: ошибка: присваивание, используемое как логическое выражение

```
if (x = 0) {
```

^

Сборка мусора



Проблемы:

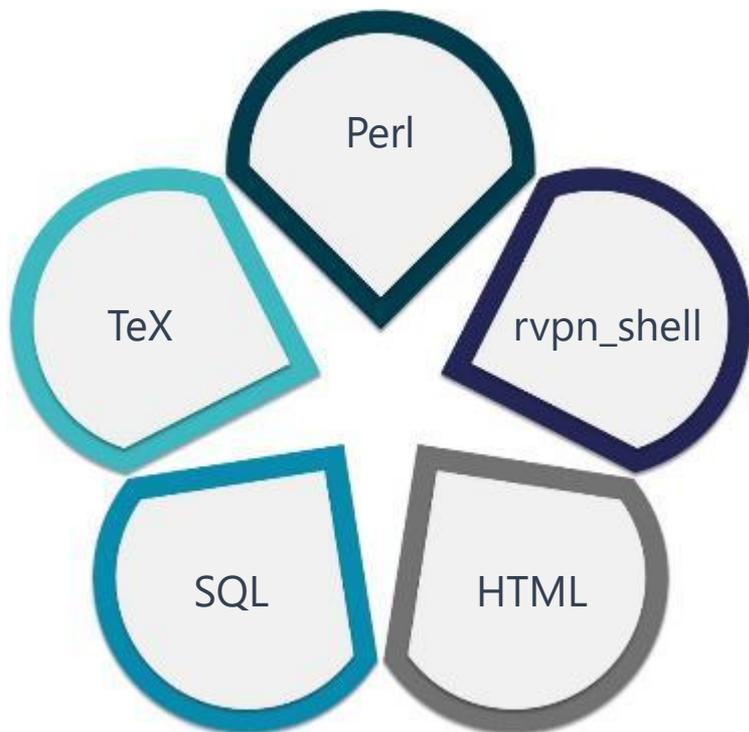
- **Висячая ссылка** – это оставшаяся в использовании ссылка на удалённый объект;
- **Утечка памяти** – сохранение неиспользуемого объекта в динамической памяти.

Сборка мусора

Обязанность освобождения памяти от объектов, которые больше не используются, возлагается на среду исполнения программы.

«**Сборщик мусора**» периодически запускается, находит неиспользуемые объекты в динамической памяти и освобождает занимаемую ими память.

Языково-ориентированная методология



Разбиение процесса разработки программного обеспечения на две стадии:

- Создание предметно-ориентированных языков (DSL);
- Описание собственно решения задачи с использованием DSL;

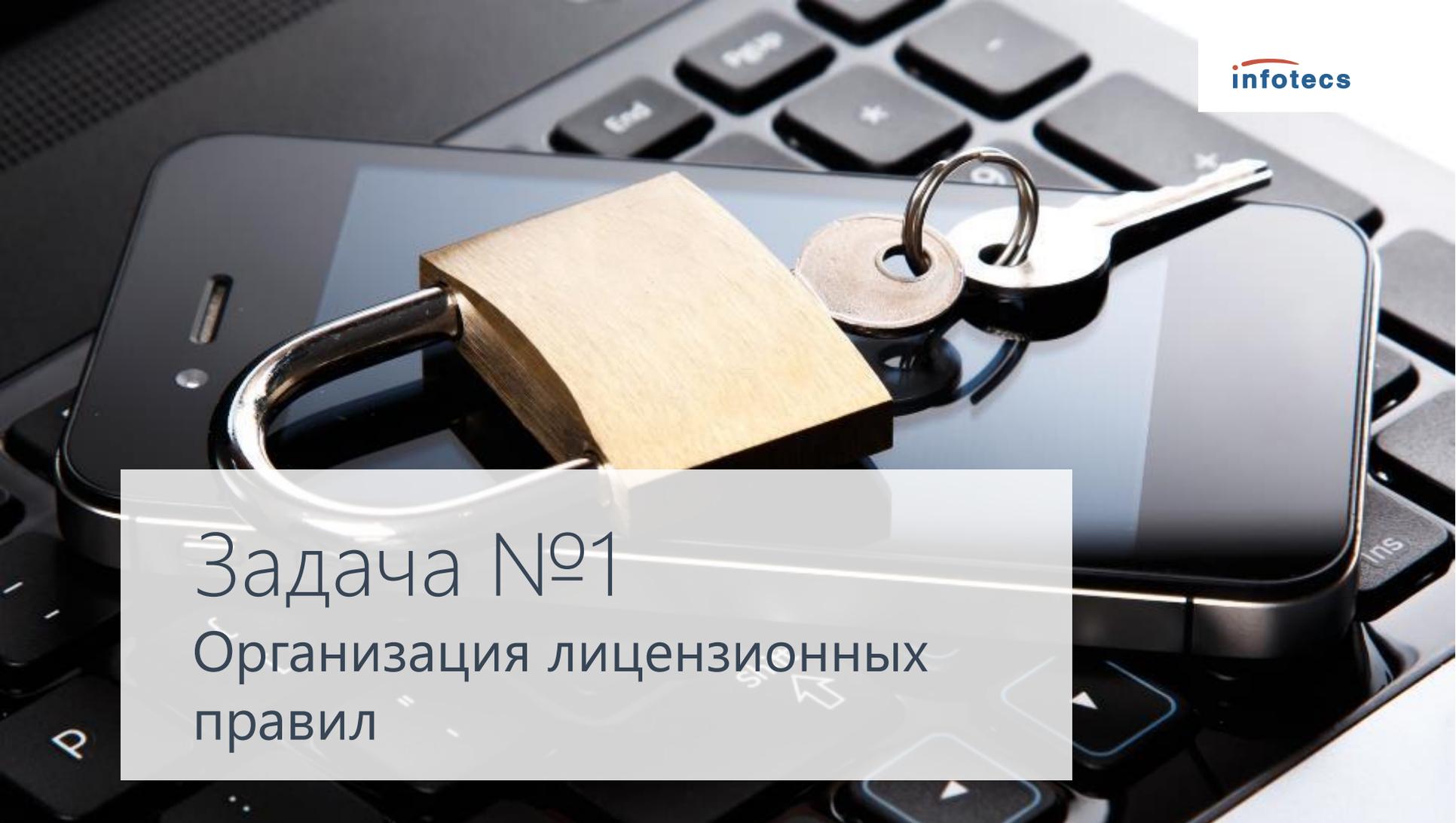
Языково-ориентированная методология



Можно создать такой язык DSL, в котором многие ошибочные с точки зрения предметной области конструкции будут просто запрещены компилятором или интерпретатором.

Технологии для исключения ненамеренных ошибок в ПО



The background of the slide is a close-up photograph of a black smartphone lying on a black laptop keyboard. A large, square wooden padlock is attached to the phone's screen. A set of keys, including a silver key and a circular metal token, is attached to the padlock's shackle. The lighting is dramatic, highlighting the textures of the wood, metal, and plastic.

Задача №1

Организация лицензионных правил

Система лицензирования ПАК ViPNet Coordinator HW



Решает комплекс задач:

- Идентификация аппаратной платформы.
- Определение свойств различных лицензионных объектов с точки зрения функциональности и производительности.
- Определение лицензионных объектов на основании справочно-ключевой информации.
- Определение соотношений между аппаратными платформами и лицензионными объектами.

Система лицензирования ПАК ViPNet Coordinator HW

Лицензионный объект разрешает определенную функциональность на компьютере с ПО ViPNet. Пример - coordinator-hw-1000.



Свойства лицензионного объекта детализируют особенности лицензионных объектов и задают некоторые настройки функционирования ПО.

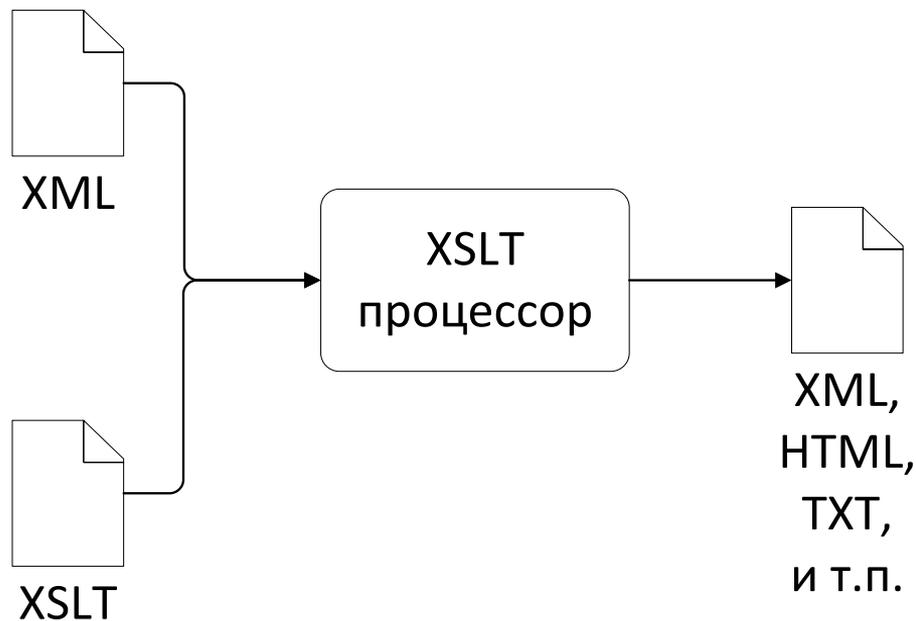
Свойства лицензионных объектов

Имя	Значение (по умолчанию)	Описание
iplir-threads	целое число ≥ 0 (0)	Максимальное число потоков шифрования. Если равно нулю, то равно числу ядер процессора.
traffic-shaper	целое число ≥ 0 (0)	Ограничение трафика в Mb/s.* Если равно нулю, то неограниченно.
default-max-connections	целое число ≥ 0 (150000)	Максимальное число соединений межсетевого экрана по умолчанию.
manual-max-connections	целое число ≥ 0 (150000)	Максимальное число соединений межсетевого экрана, устанавливаемое вручную.

Проблема

Лицензионные правила кодировались в скриптах или в компилируемом коде программ. Это неудобно с точки зрения локализации изменений, обзорности и четкости понимания правил по коду. Данный способ реализации правил лицензирования приводил к **ошибкам**.

Технология XSLT-преобразований



Декларация лицензионных правил в формате XML

```
<liclimits>
  <group comment="Russian hw1000">
    <licobjects>
      <licobject id="coordinator-hw-1000"/>
    </licobjects>
    <properties>
      <property name="iplir-threads" value="2"/>
      <property name="failover" value="yes"/>
      <property name="default-max-connections" value="800000"/>
      <property name="manual-max-connections" value="1000000"/>
    </properties>
  </group>
</liclimits>
```

Трансляция лицензионных деклараций в исходный код

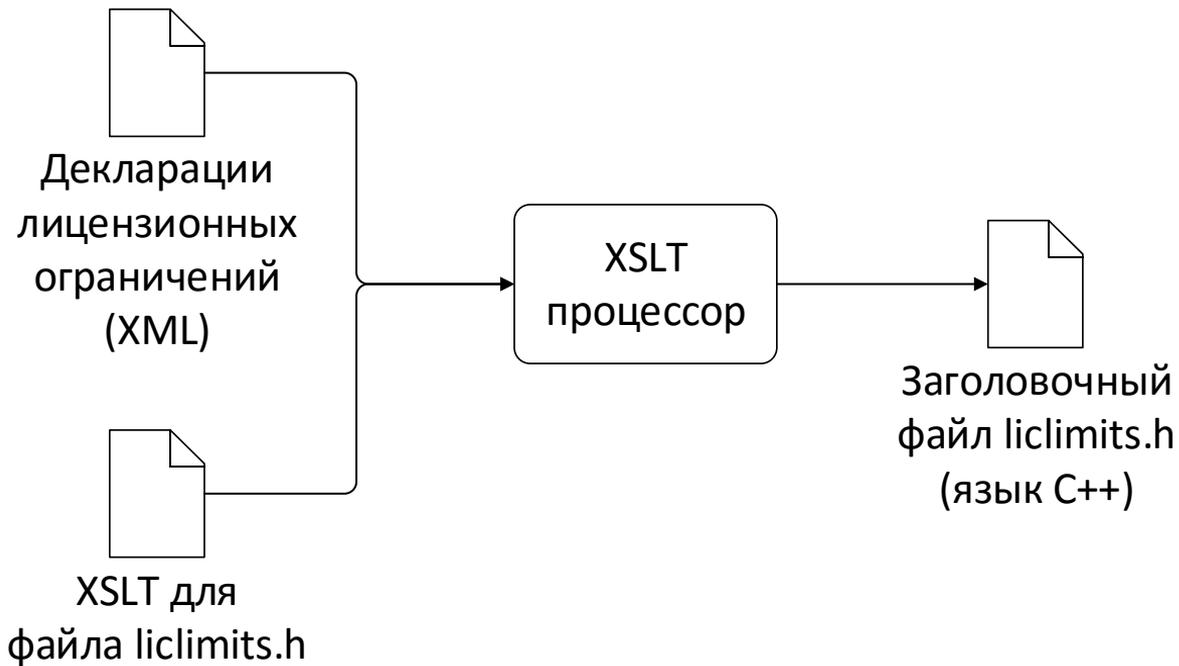


Таблица стилей XSLT

```
<xsl:for-each select="group">
  <xsl:for-each select="licobjects">
    <xsl:for-each select="licobject">
      <xsl:variable name="licobjid" select="@id"/>
      <xsl:for-each select="../../../properties/property">
        <xsl:text>      result["</xsl:text>
        <xsl:value-of select="$licobjid"/>
        <xsl:text>"] ["</xsl:text>
        <xsl:value-of select="@name"/>
        <xsl:text>"] = "</xsl:text>
        <xsl:value-of select="@value"/>
        <xsl:text>";
      </xsl:for-each>
    </xsl:for-each>
  </xsl:for-each>
</xsl:for-each>
```

Таблица стилей XSLT

```
<xsl:for-each select="group">
  <xsl:for-each select="licobjects">
    <xsl:for-each select="licobject">
      <xsl:variable name="licobjid" select="@id"/>
      <xsl:for-each select="../../../properties/property">
        <xsl:text>      result["</xsl:text>
        <xsl:value-of select="$licobjid"/>
        <xsl:text>"] ["</xsl:text>
        <xsl:value-of select="@name"/>
        <xsl:text>"] = "</xsl:text>
        <xsl:value-of select="@value"/>
        <xsl:text>";
      </xsl:for-each>
    </xsl:for-each>
  </xsl:for-each>
</xsl:for-each>
```

Таблица стилей XSLT

```
<xsl:for-each select="group">
  <xsl:for-each select="licobjects">
    <xsl:for-each select="licobject">
      <xsl:variable name="licobjid" select="@id"/>
      <xsl:for-each select="../../properties/property">
        <xsl:text>      result["</xsl:text>
        <xsl:value-of select="$licobjid"/>
        <xsl:text>"] ["</xsl:text>
        <xsl:value-of select="@name"/>
        <xsl:text>"] = "</xsl:text>
        <xsl:value-of select="@value"/>
        <xsl:text>";
      </xsl:for-each>
    </xsl:for-each>
  </xsl:for-each>
</xsl:for-each>
```

Таблица стилей XSLT

```
<xsl:for-each select="group">
  <xsl:for-each select="licobjects">
    <xsl:for-each select="licobject">
      <xsl:variable name="licobjid" select="@id"/>
      <xsl:for-each select="../../properties/property">
        <xsl:text>      result["</xsl:text>
        <xsl:value-of select="$licobjid"/>
        <xsl:text>"] ["</xsl:text>
        <xsl:value-of select="@name"/>
        <xsl:text>"] = "</xsl:text>
        <xsl:value-of select="@value"/>
        <xsl:text>";
      </xsl:for-each>
    </xsl:for-each>
  </xsl:for-each>
</xsl:for-each>
```

Таблица стилей XSLT

```
<xsl:for-each select="group">
  <xsl:for-each select="licobjects">
    <xsl:for-each select="licobject">
      <xsl:variable name="licobjid" select="@id"/>
      <xsl:for-each select="../../properties/property">
        <xsl:text>      result["</xsl:text>
        <xsl:value-of select="$licobjid"/>
        <xsl:text>"] ["</xsl:text>
        <xsl:value-of select="@name"/>
        <xsl:text>"] = "</xsl:text>
        <xsl:value-of select="@value"/>
        <xsl:text>";
      </xsl:for-each>
    </xsl:for-each>
  </xsl:for-each>
</xsl:for-each>
```

Сгенерированный заголовочный файл (C++)

```
static Limits
liclimits_create_limits( void )
{
    Limits result;
    result["coordinator-hw-1000"]["iplir-threads"] = "2";
    result["coordinator-hw-1000"]["failover"] = "yes";
    result["coordinator-hw-1000"]["default_max_connections"] = "800000";
    result["coordinator-hw-1000"]["manual_max_connections"] = "1000000";
    return result;
}
```

Интерфейс для получения лицензионных деклараций

```
namespace hwlic {  
    // Defines liclimits_create_limits  
    #include <hwlic/module/liclimits.h>  
}  
  
namespace {  
    const Limits liclimits = hwlic::liclimits_create_limits();  
}
```

Интерфейс для получения лицензионных деклараций

```
std::map<std::string, std::string>
licobject_props( const std::string& licobjid )
{
    if ( liclimits.count( licobjid ) == 0 )
    {
        throw std::runtime_error( "The specified licobject in unknown." );
    }

    return liclimits.find(licobjid)->second;
}
```

The background of the slide is a top-down photograph of a group of business professionals sitting around a large, light-colored wooden conference table. Several individuals are using mobile devices: one man uses a laptop, another uses a tablet, and several others use smartphones. The scene is brightly lit, suggesting a modern office environment.

Задача №2

автоматизация настройки ПАК

Настройка ПАК через CLI

Пример настройки статического адреса на сетевом интерфейсе:

```
inet ifconfig eth2 address 10.0.88.1 netmask 255.255.255.0  
inet ifconfig eth2 up
```

Настройка ПАК через CLI

Пример настройки статического адреса на сетевом интерфейсе:

```
inet ifconfig eth2 address 10.0.88.1 netmask 255.255.255.0  
inet ifconfig eth2 up
```

Метод неудобен для решения следующих задач:

- Первичная настройка большого числа ПАК-ов одинаковым образом.
- Внесении одинаковых изменений в конфигурации уже функционирующих ПАК-ов.

Централизованная настройка ПАК

Производится через передачу на ПАК файла в формате XML:

```
<networking>  
  <interfaces>  
    <interface name="eth2">  
      <param name="enable" value="yes"/>  
      <param name="mode" value="static"/>  
      <param name="ip_mask" value="10.0.88.1/24"/>  
    </interface>  
  </interfaces>  
  <routes/>  
</networking>
```

Трансляция настроек ПАК в команды CLI

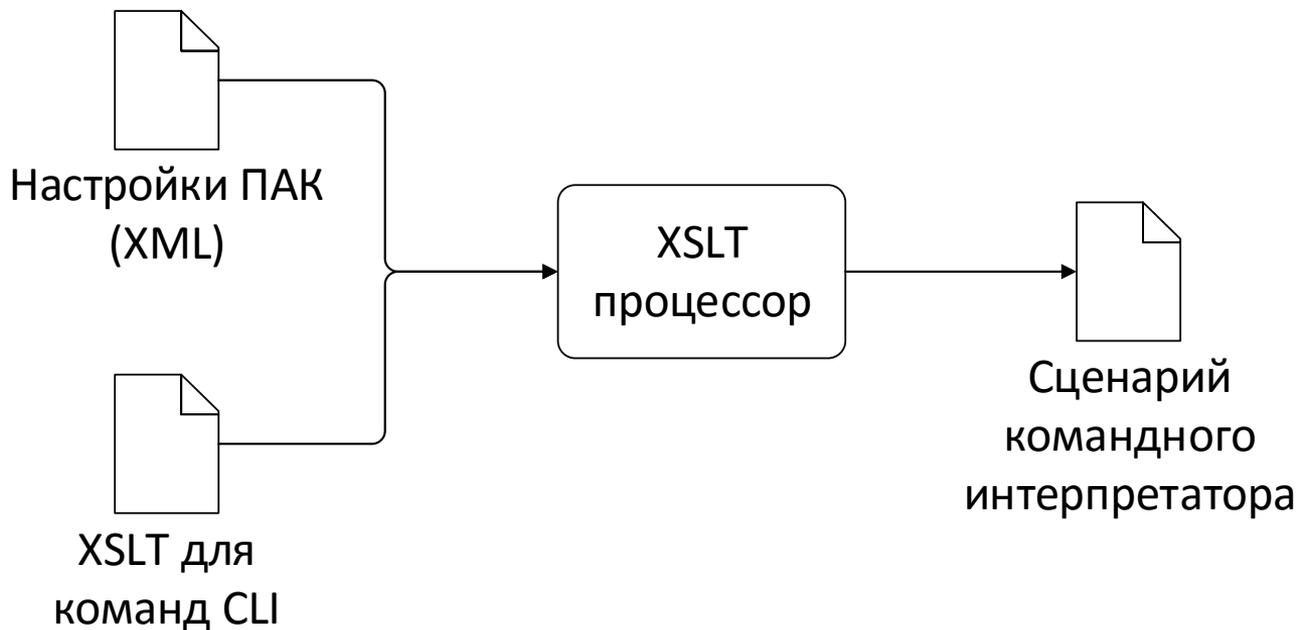
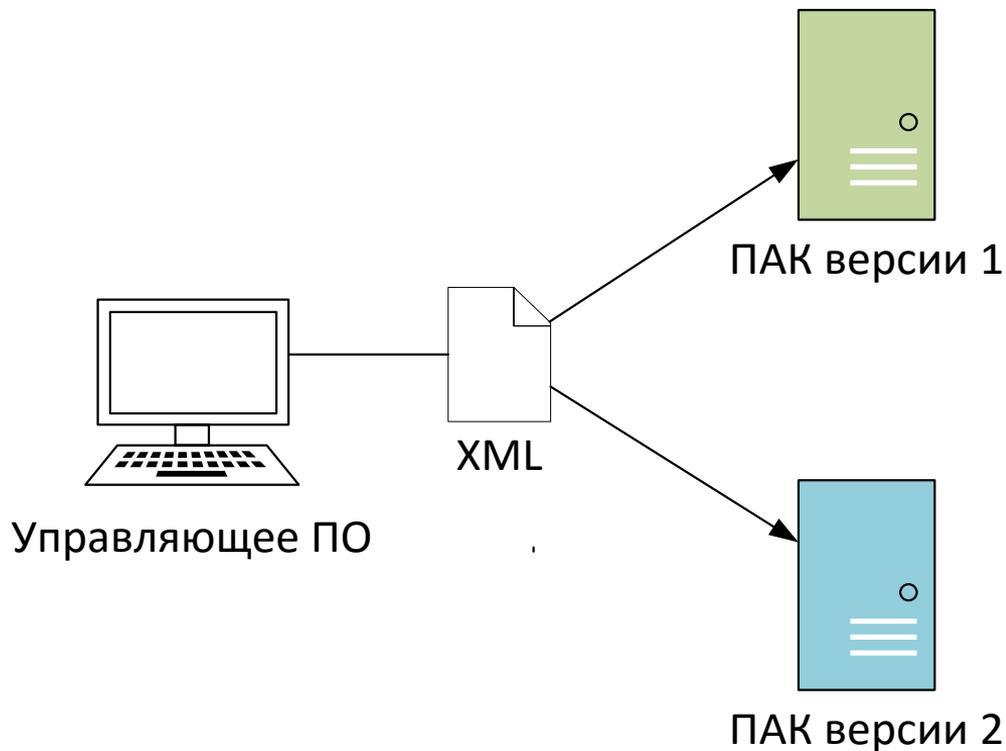


Таблица стилей XSLT

```
<xsl:if test="@name = 'ip_mask'">
  <xsl:text>inet ifconfig </xsl:text>
  <xsl:value-of select="../@name"/>
  <xsl:text> address </xsl:text>
  <xsl:value-of select="substring-before(@value, '/')"/>
  <xsl:text> netmask </xsl:text>
  <xsl:call-template name="mask2mask">
    <xsl:with-param name="mask" select="substring-after(@value, '/')"/>
  </xsl:call-template>
  <xsl:text>
</xsl:text>
</xsl:if>
```

Упрощение совместимости с управляющим ПО



Упрощение совместимости разных версий ПО



**ViPNet Security
Gateway 3.2**

Реализация IPSec

Racoon



**ViPNet Coordinator
HW 4.1**

StrongSwan

Упрощение совместимости разных версий ПО



**ViPNet Security
Gateway 3.2**

Реализация IPSec

Racoon



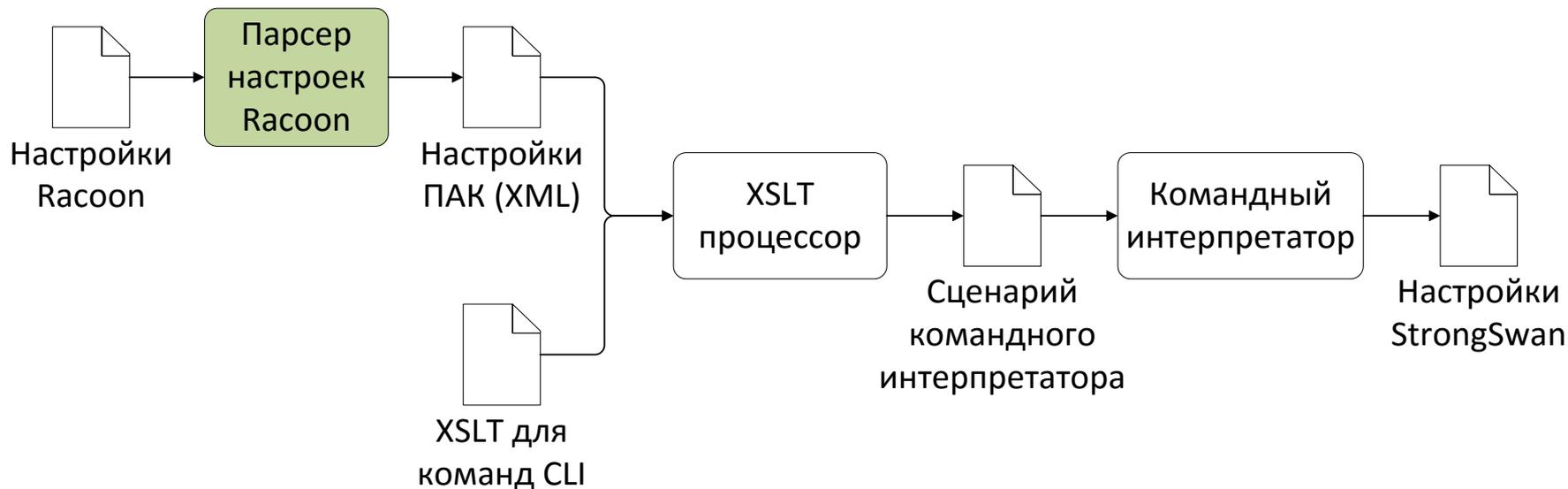
**ViPNet Coordinator
HW 4.1**

StrongSwan

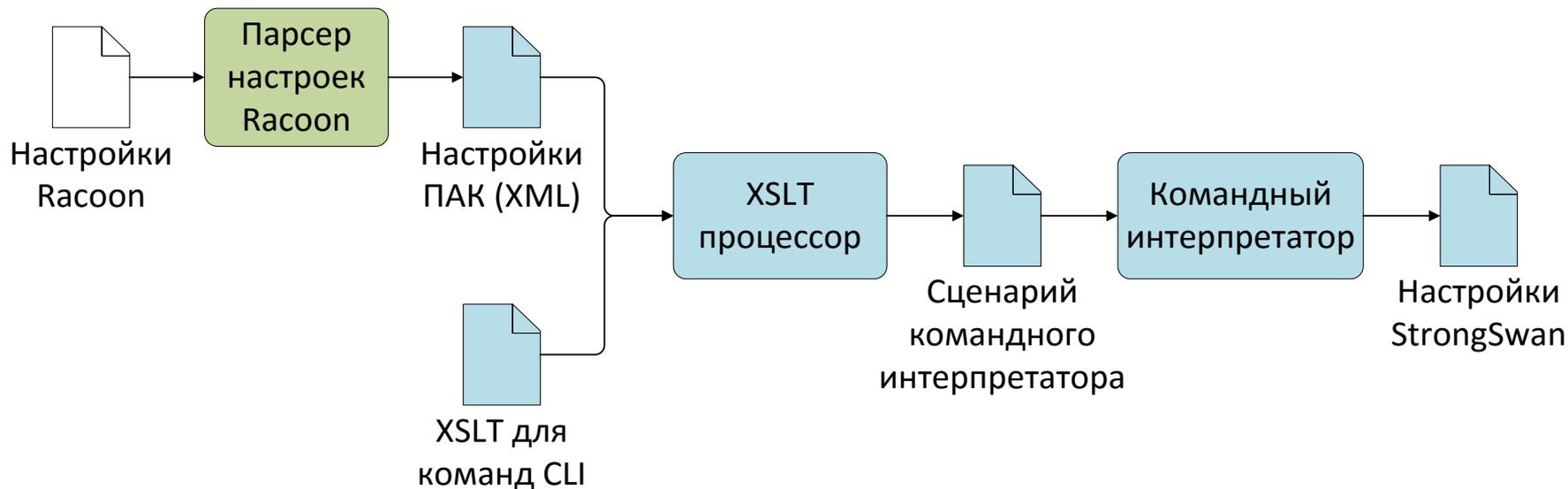
Задача:

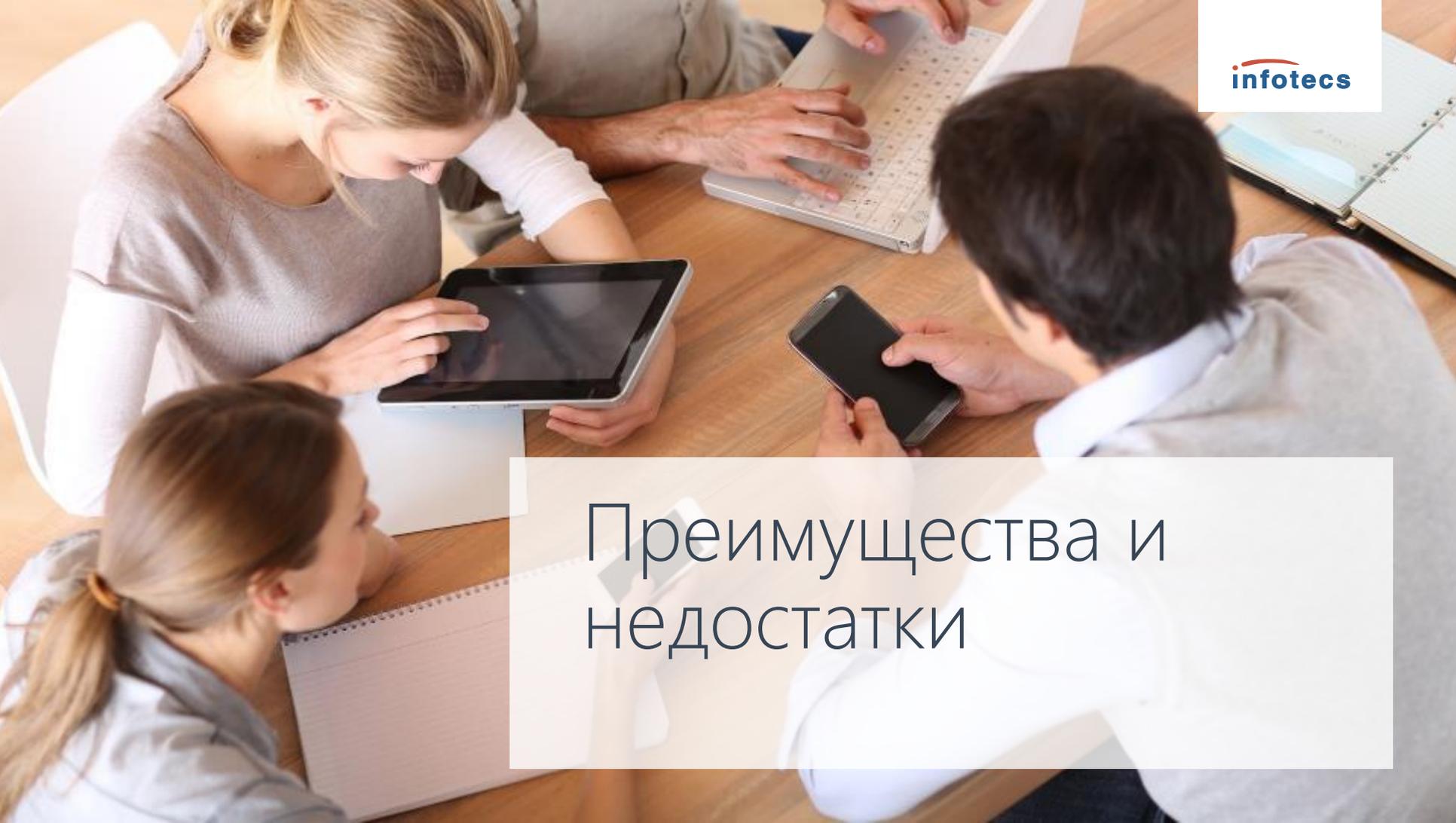
Обеспечить конвертацию настроек Racoon в настройки StrongSwan без жёсткой привязки к текущей реализации IPSec.

Упрощение совместимости разных версий ПО



Упрощение совместимости разных версий ПО



A high-angle photograph of four people in a meeting. A woman on the left uses a tablet, a man on the right uses a smartphone, and another person in the background uses a laptop. A woman in the foreground is looking at a notebook. A semi-transparent white box with text is overlaid on the bottom right.

Преимущества и недостатки

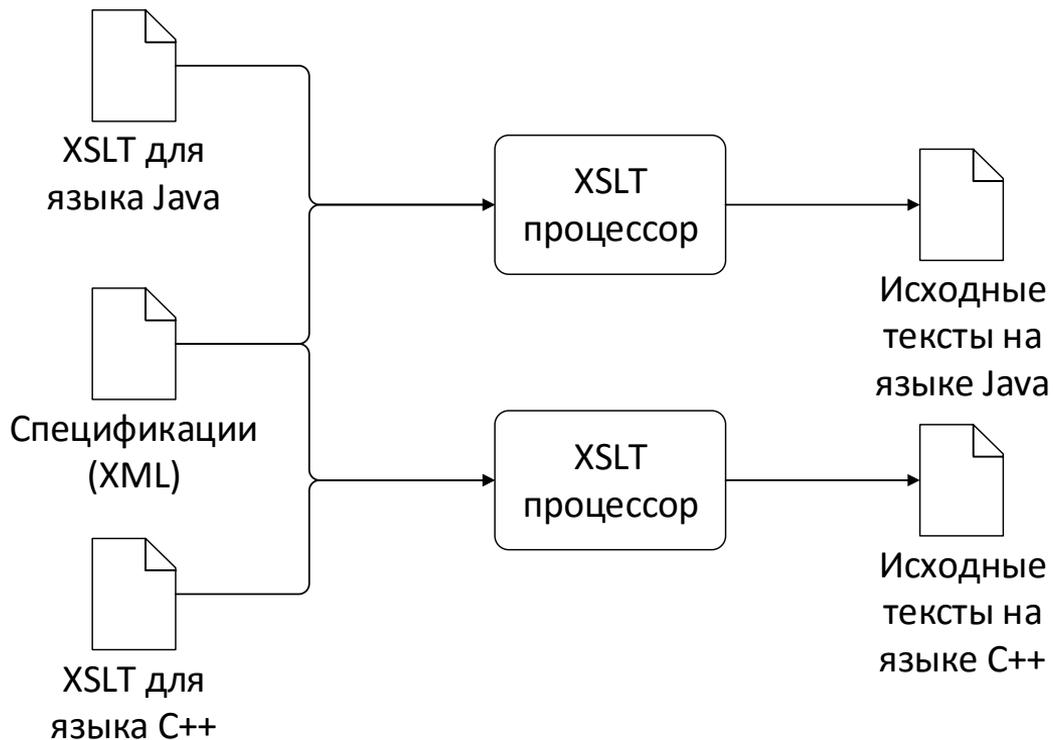
Преимущества

- + Возможность для управления спецификациями ПО непосредственно аналитиком минуя программиста.

Преимущества

- + Возможность для управления спецификациями ПО непосредственно аналитиком минуя программиста.
- + Упрощение обеспечения совместимости различных версий ПО.

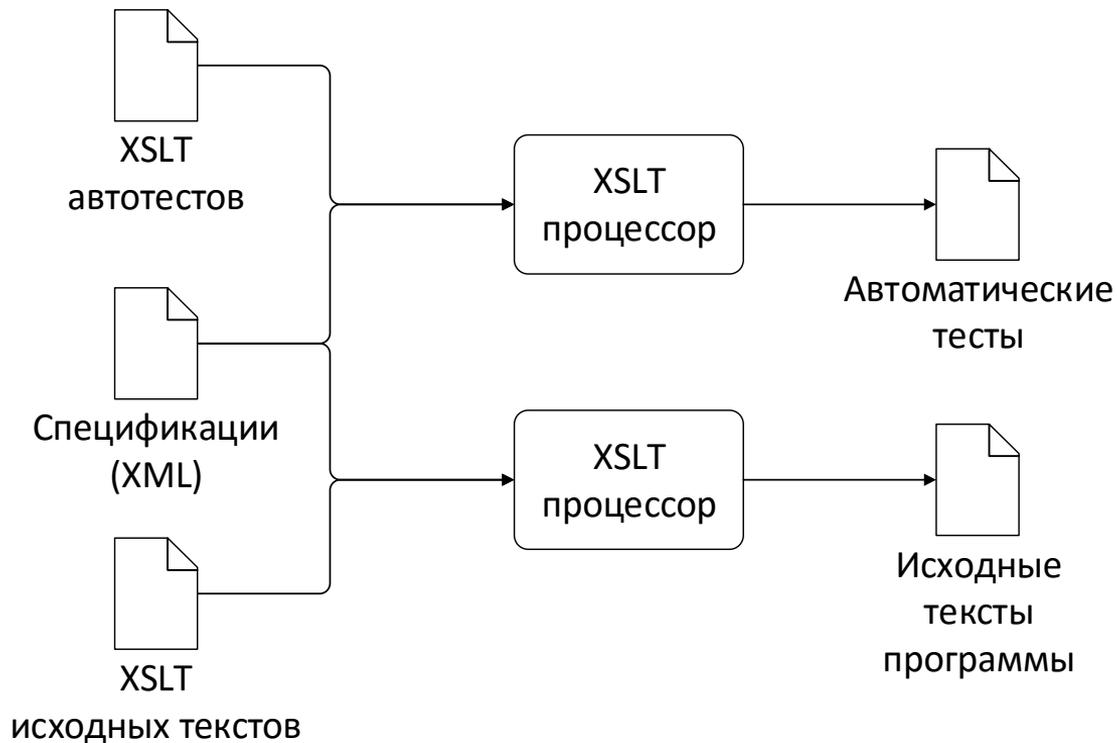
Генерация исходных текстов на разных языках



Преимущества

- + Возможность для управления спецификациями ПО непосредственно аналитиком минуя программиста.
- + Упрощение обеспечения совместимости различных версий ПО.
- + На основе спецификаций можно генерировать не только исходный код, но и автоматические тесты ПО.

Генерация автоматических тестов



Преимущества

- + Возможность для управления спецификациями ПО непосредственно аналитиком минуя программиста.
- + Упрощение обеспечения совместимости различных версий ПО.
- + На основе спецификаций можно генерировать не только исходный код, но и автоматические тесты ПО.
- + Сокращает сроки разработки за счёт упрощения корректировки спецификаций.

Недостатки



- Подходит только для некоторых хорошо формализуемых задач.

Недостатки

- Подходит только для некоторых хорошо формализуемых задач.
- Усложняется процедура сборки ПО как следствие добавления новой стадии – трансляции XML-деклараций в исходный код.

Недостатки

- Подходит только для некоторых хорошо формализуемых задач.
- Усложняется процедура сборки ПО как следствие добавления новой стадии – трансляции XML-деклараций в исходный код.
- Появляются новые типы потенциальных ошибок, которые нужно учитывать при реализации.

The background of the slide is a photograph of several wind turbines silhouetted against a bright, orange, and cloudy sunset sky. The turbines are positioned on the left and right sides of the frame, with a power line tower visible in the distance.

Спасибо!
Вопросы?

Минко В.С.
системный архитектор
vitaly.minko@infotecs.ru

ИнфоТеКС
www.infotecs.ru

ИнфоТеКС
Академия
academy.infotecs.ru

Перспективный
мониторинг
amonitoring.ru