От открытого кода к отечественным сертифицированным системам

Владимир Рубанов
Президент и генеральный конструктор
ООО «НТЦ ИТ РОСА»

Требования к импортозамещению ПО

Что делать? Варианты.

1. Использовать готовое **зарубежное ПО**.

2. Создавать **отечественное ПО «с нуля»** полностью силами российских специалистов.

3. Создавать **отечественное ПО** на основе доработки **открытых и свободных международных компонентов (open-source)**.

1. Используем зарубежное ПО

Ничего не нужно делать, самый легкий вариант

Негативные последствия могут включать:

- ✓ Отключение поддержки и обновлений производителя (в т.ч. по безопасности)
- ✓ Невозможность самостоятельной доработки и исправлений
- ✓ Отзыв лицензий на основании пункта договоров о необходимости соблюдения законов государства юрисдикции производителя
- ✓ Удаленную активацию встроенных закладок – от утечек данных до нарушения функциональности работы.

2. Разрабатываем полностью российское «с нуля»

- Самый полный контроль
- Дорого. Очень дорого
- Долго. Очень долго
- Нет достаточного количества специалистов
- Изоляция от зарубежного сообщества снижает качество

Там, где российское уже есть (1С, ABBYY, Kaspersky и т.д.) – конечно используем!

3. Адаптируем и дорабатываем СПО (open-source) - 1

- Легитимный способ воспользоваться готовыми наработками **многих тысяч человеко-лет** мирового прогресса. **Бесплатно**.
- **Свобода использования и разработки** – возможность самостоятельно дорабатывать, исправлять ошибки, выпускать разные варианты исполнения.
- Наличие *полных исходных кодов и средств разработки* дает больше возможностей **контроля безопасности**, в т.ч. при сертификации.

3. Адаптируем и дорабатываем СПО (open-source) - 2

В «мирное время» сохраняется **связь и синергия с зарубежными разработчиками**. Разработка общих частей идет совместными усилиями. Затраты делятся между участниками, а результат принадлежит всем в равной степени.

Нельзя полностью диктовать **зарубежным коллегам** что, как и когда делать. Хотя в любой момент можно сделать **собственную модифицированную копию**.

Открытое не означает автоматически **безопасное**. Необходимы проверки.

Технологическая независимость не означает изоляцию

Технологическая независимость заключается не в изоляции от мирового сообщества в «мирное время», а в потенциальной способности страны обеспечить функционирование текущих ИТ-систем и продолжить их развитие силами локальных специалистов в гипотетических условиях форс-мажора

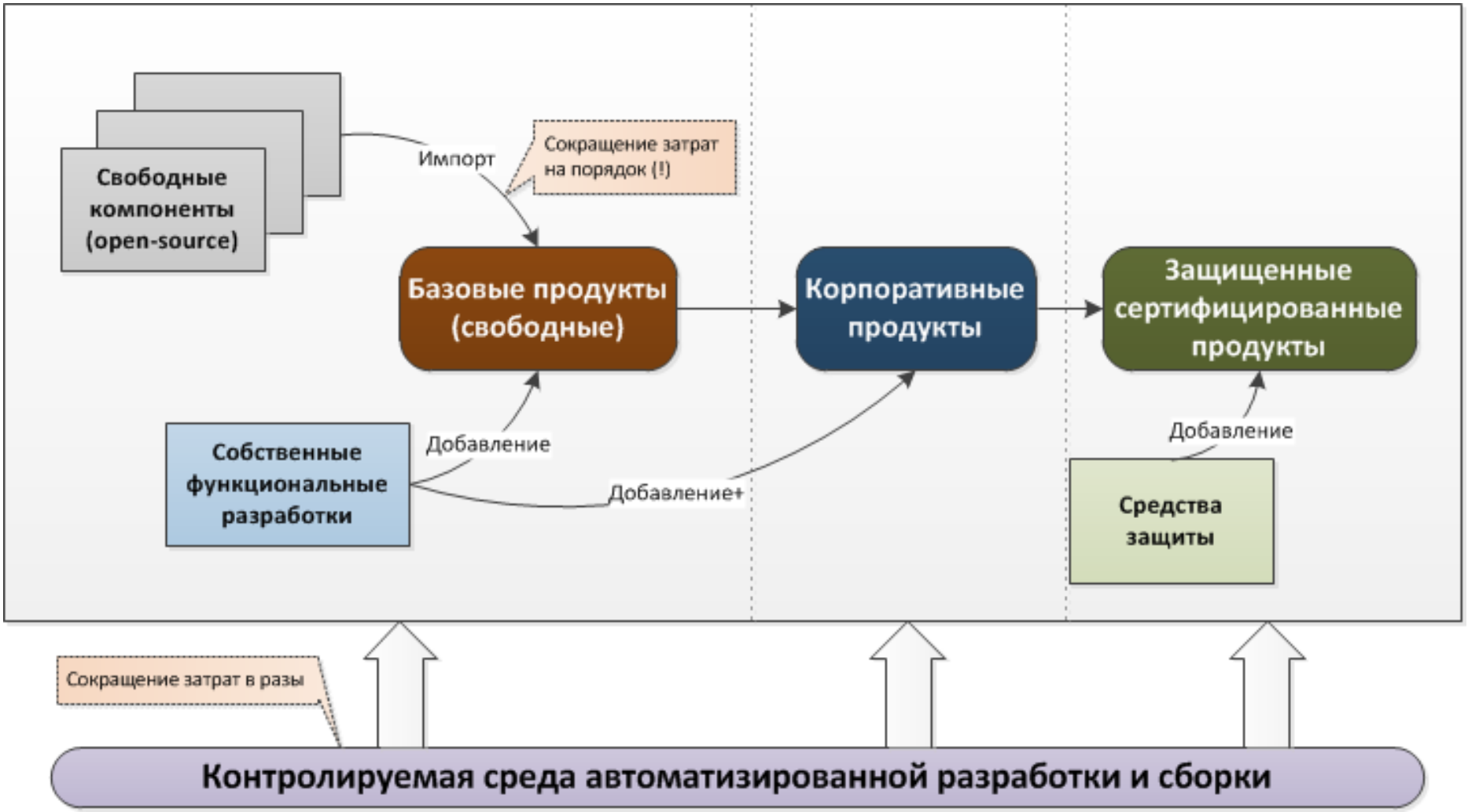
Иметь возможность – не значит применять!

Составляющие технологической независимости ПО

Наличие внутри страны

1. **Исходные коды**
2. **Средства разработки и сборки**
3. **Локальные специалисты** по поддержке, доработке функционала, исправлению ошибок.

Совмещаем свободу и контроль



Семейство ОС РОСА

Тип задач	Серверные	Настольные
Домашние / community		ROSA Fresh
Корпоративные незащищенные	ROSA Enterprise Linux Server	ROSA Enterprise Desktop
Обработка конфиденциальной информации и персональных данных (1Г)	РОСА Кобальт SX	РОСА Кобальт DX
Обработка информации, содержащей государственную тайну (1В)	РОСА Хром SX	РОСА Хром DX

Совместимость с отечественными разработками

Выводы (1)

1. В отдельных некритичных областях можно пока использовать зарубежное ПО.
   - ✓ Важно: обеспечивать **совместимость** с отечественным и свободным стеком на основе открытых стандартов.
2. В тех сегментах, где есть 100% отечественное ПО – ему «зеленый свет».
   - ✓ Важно: не допускать **монополий** – в каждом сегменте как минимум 2 конкурирующих производителя.

Выводы (2)

3. В прочих сегментах – брать за основу **открытые (open-source) компоненты**, дорабатывать их и строить на их основе **отечественные продукты**, поддержку и развитие которых могут обеспечить российские компании.

- ✓ Важно: сохранять международную кооперацию в «мирное время».
- ✓ Важно: в любой момент иметь возможность перейти в «автономный режим».
- ✓ Важно: по мере роста ресурсов и компетенций в определенных сегментах увеличивать долю собственного кода – в отдельных случаях переходя в 100% .

Владимир Рубанов

Президент и
генеральный конструктор

к.ф.-м.н., PMP

rubanov@rosalab.ru

http://rosalab.ru/
http://rubanov.pro/

Открытость в разработке ПО

- Технологическая открытость :
  - ✓ Открытый код
    - свобода разработки программ
  - ✓ Открытые данные / форматы
    – свобода обмена данными
  - ✓ Открытые API / протоколы
    – свобода взаимодействия программ

- Открытость профессиональной среды:
  - ✓ Открытые проекты разработки
  - ✓ Открытые сообщества и консорциумы
  - ✓ Открытые методы обучения специалистов

Миф 1: свободные лицензии не совместимы с бизнесом

- Свободные лицензии:
  1. Copyleft : GPL, LGPL и др.
  2. Non-copyleft (permissive): BSD, MIT, Apache и др.
- На самом деле:
  - ✓ НЕ запрещают <u>продавать</u> продукты на основе свободных компонентов
  - ✓ НЕ обязывают <u>распространять</u> код широковещательно
  - ✓ Permissive лицензии позволяют строить полностью проприетарные продукты

Миф 2: открытость ПО автоматически дает безопасность

- Открытый код сам по себе не означает безопасное ПО.
- Однако, открытость делает *возможным*:
  - ✓ Более глубокий анализ на уязвимости и недекларированные возможности (с помощью автоматизированных средств!).
  - ✓ Независимую от производителя доработку и исправление ошибок и уязвимостей.