



Прогрессивные технологии для интеллектуальных систем



QNX для производителей  
телекоммуникационного оборудования  
Технологии и лицензионная политика

Александр Трофимов

Менеджер по продажам отраслевых решений

# О чем пойдет речь

- **Выбор платформы - вопрос стратегии**
- **Критические точки при выборе ОС:**
  - > **Надежность**
  - > **Масштабируемость**
  - > **Минимизация рисков**
- **SWD Software – технологический партнер телекоммуникационных компаний**



Вопрос выбора платформы –  
вопрос стратегии

# Особенности рынка телекоммуникаций

## → Большие тиражи

- > Необходимость гарантийного и послепродажного сопровождения
- > Большие первоначальные вложения
- > Серьезные финансовые риски в случае неудачи

## → Быстрая смена технологий и жесткая конкурентная борьба

- > Производитель вынужден развиваться в условиях большой неопределенности
- > Критично время разработки или добавления функциональности

## → Выпуск одновременно нескольких продуктовых линеек

- > Необходимо оптимизировать затраты на разработку всех линеек

# ПО часто не уделяют должного внимания

- Лишь 10% сбоев ИТ-оборудования вызвано аппаратными отказами
- ПО определяет потребительские свойства изделия в целом:
  - > Надежность устройства
  - > Функциональность устройства
  - > Удобство использования устройства
- ПО меняется значительно чаще аппаратуры
  - > Разные прошивки – разные устройства



Источник: Gartner Group, 2001

# Критические точки при выборе ОС

- **Надежность**
- **Масштабируемость**
- **Минимизация рисков**



# Надежность

# Что если система недостаточно надежна?

- Не подходит для оборудования операторского класса
  - > Требуется работа в режиме высокой готовности 24\*7
  - > Требуется постоянная поддержка с выездом инженера в случае сбоев
- Огромные расходы на послепродажное обслуживание и поддержку
  - > Больше тираж – больше обращений
  - > Большая группа поддержки – большие расходы компании-производителя
- Ненадежное оборудование не будут покупать пользователи
  - > В классе SOHO надежность также важна
  - > Домашним пользователям community посоветует более надежные устройства





# Высокая готовность

$$\text{Готовность} = \frac{\text{Mean Time Between Failure}}{\text{Mean Time Between Failure} + \text{Mean Time To Repair}}$$

> **99.999%** эквивалентна < 5 минутам простоя в год

> **Увеличение MTBF повышает готовность**

- MTBF ПО зависит от вероятности дефектов (кол-во дефектов на тысячу строк кода)
- MTBF увеличивается лучшими практиками качества ПО

> **Уменьшение MTTR повышает готовность**

- Если MTTR будет равна нулю, готовность будет 100%

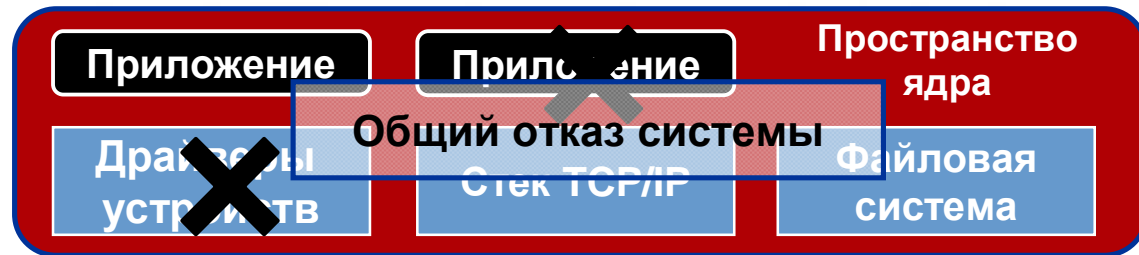
# Повышение готовности ПО

- Применить практики качества ПО для сокращения MTBF
- Использовать слабую связанность компонентов исполняемой среды для снижения риска каскадных сбоев
- Сократить MTTR программных компонентов
- Сократить последовательные зависимости
  - > Сбой одного компонента выводит из строя зависимый компонент
- По возможности дублировать программные компоненты
  - > Параллельные компоненты повышают готовность
- Сократить последствия сбоев в среде исполнения

# Архитектура ОС и готовность

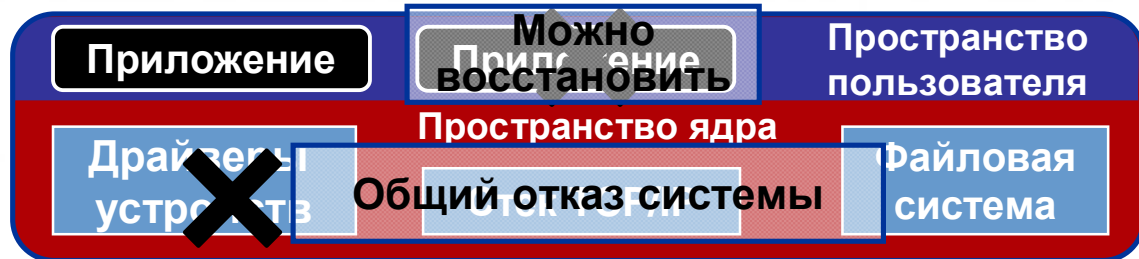
## Исполняемый модуль реального времени (VxWorks, VRTX, ThreadX)

- > Защиты памяти нет
- > Приложения, драйверы и протоколы "живут" в пространстве ядра



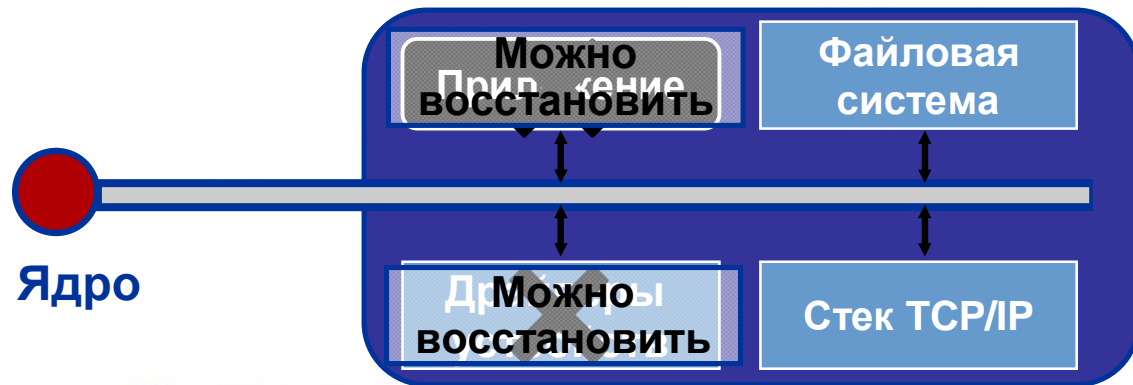
## Монолитное ядро (NT / Unix / и т.п.)

- > MMU, частичная защита
- > Защищены только приложения

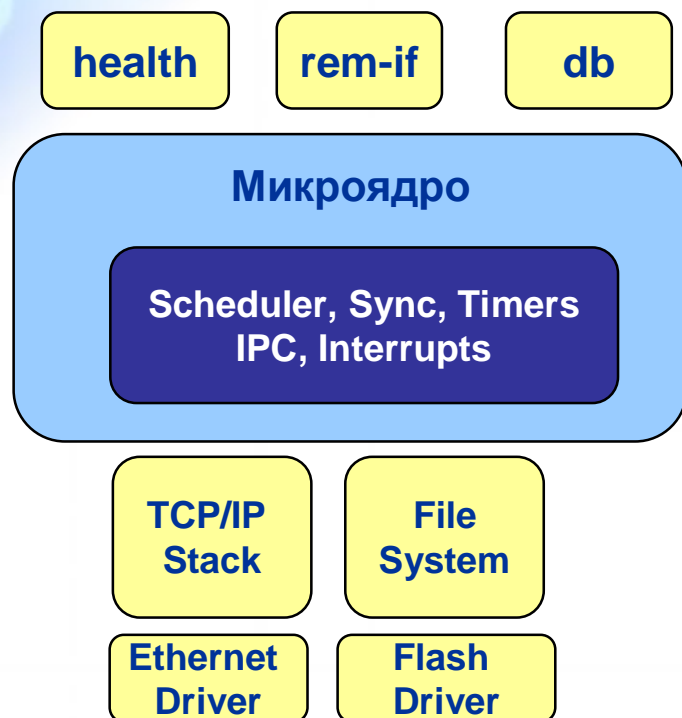


## Микроядро (QNX Neutrino)

- > MMU, полная защита
- > Защищены приложения, драйверы и протоколы

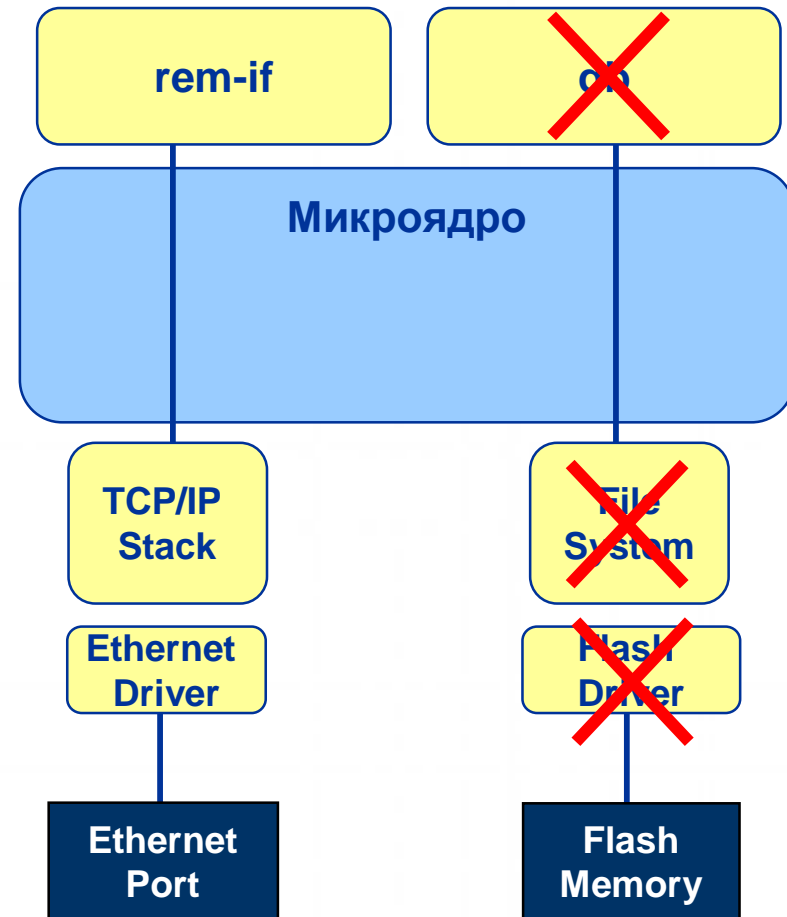
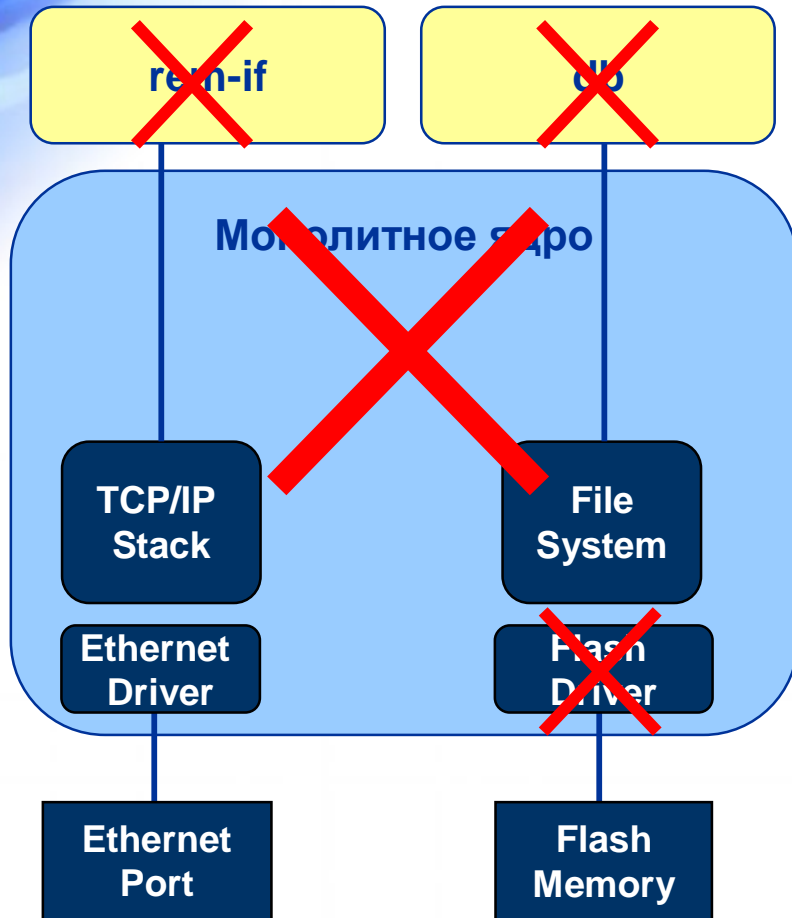


# Микроядро – основа высокой готовности



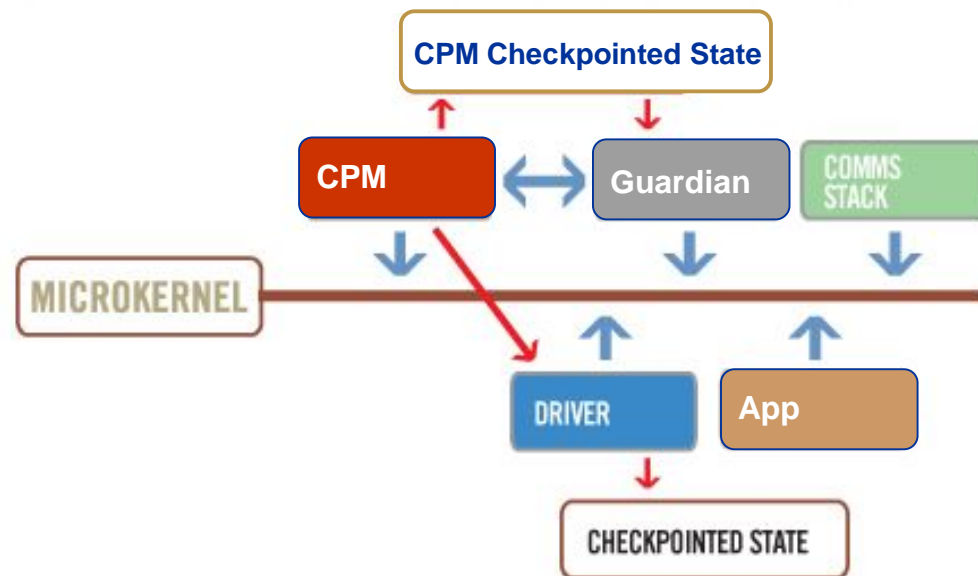
- QNX® Neutrino®
- Сокращает последовательные зависимости – сервисы ОС вынесены из ядра
- Минимальны последствия сбоя драйвера или сервиса ОС – сбой влияет только на приложение, работающее с сервисом/драйвером
- Маленькое MTTR т.к. после сбоев идет простое восстановление
- Ошибки приложений не влияют на всю систему
- Приложения могут быть индивидуально перезапущены/восстановлены

# Минимизация последствия сбоев

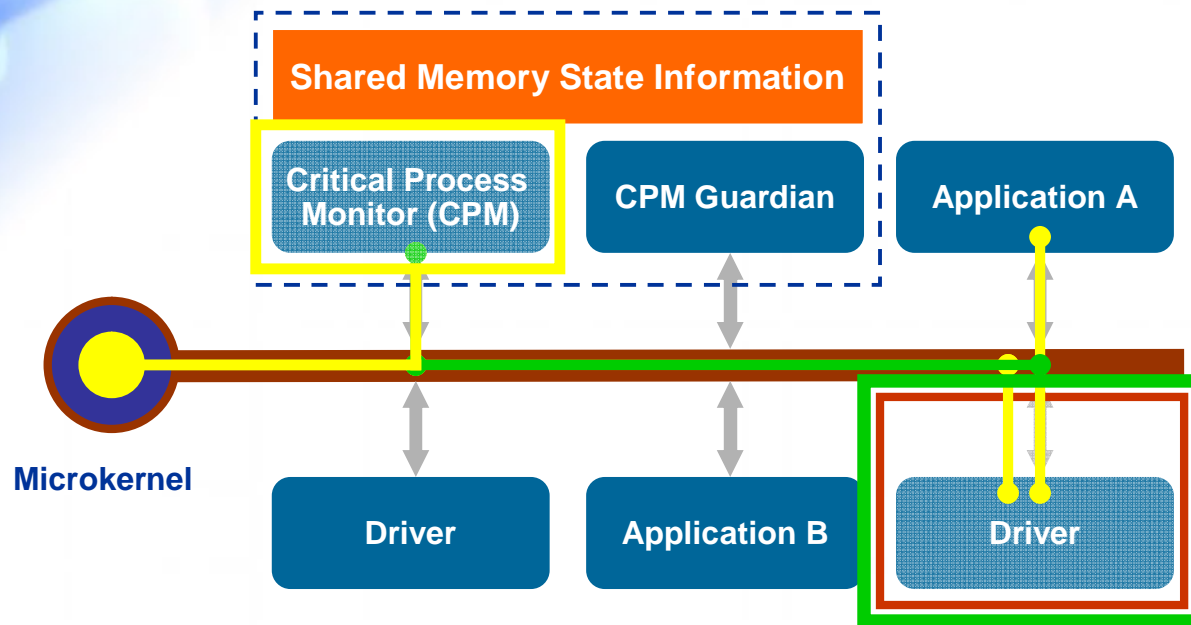


# Сокращение MTTR

- Мониторинг выбранных компонентов
- Процесс-охранник дублирует CPM
- Клиентские соединения автоматически восстанавливаются
- Конфигурируемый сценарий восстановления:
  - ▶ освободить ресурсы
  - ▶ перезапустить процесс
  - ▶ ...и т.п.

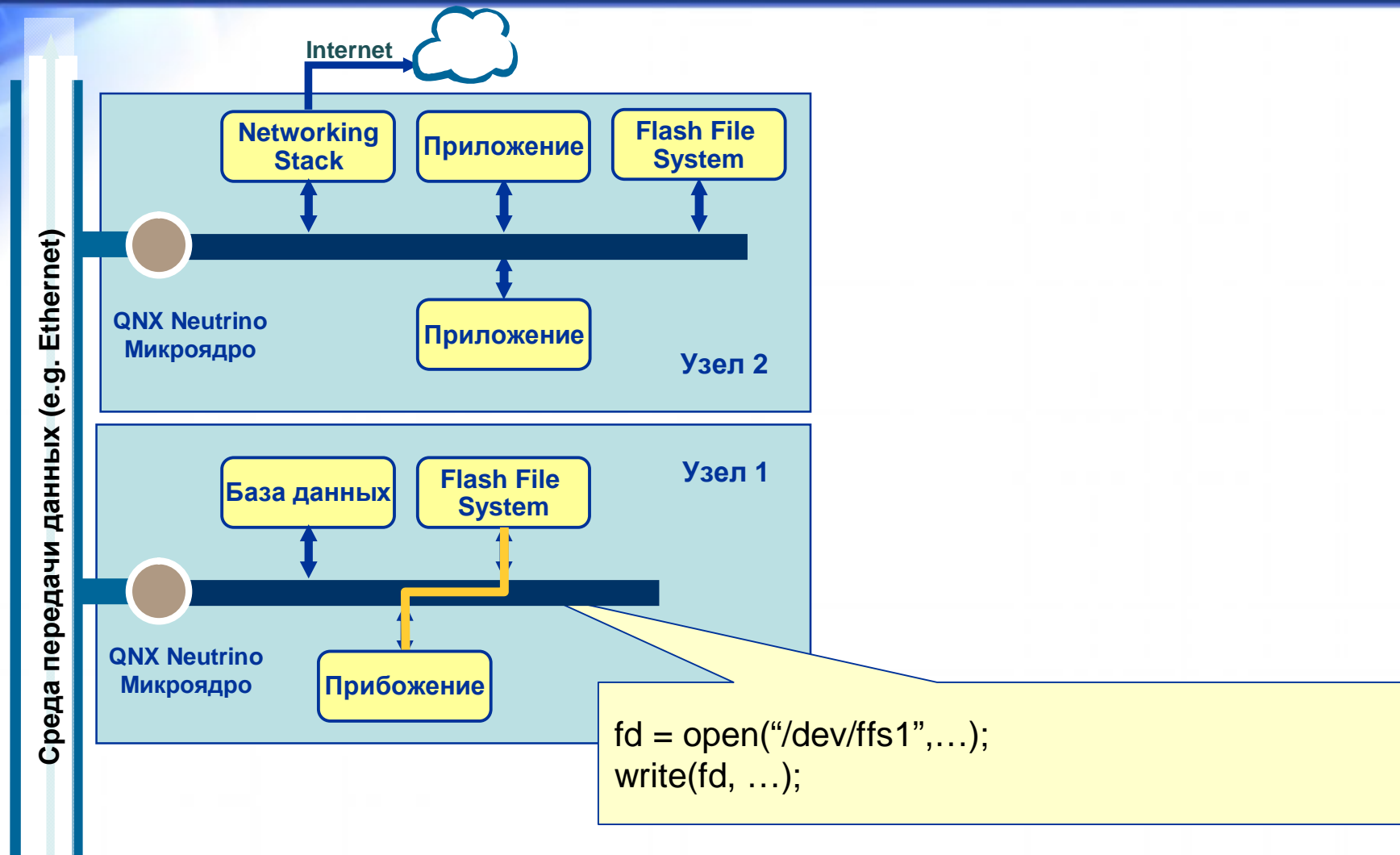


# Сокращение MTTR



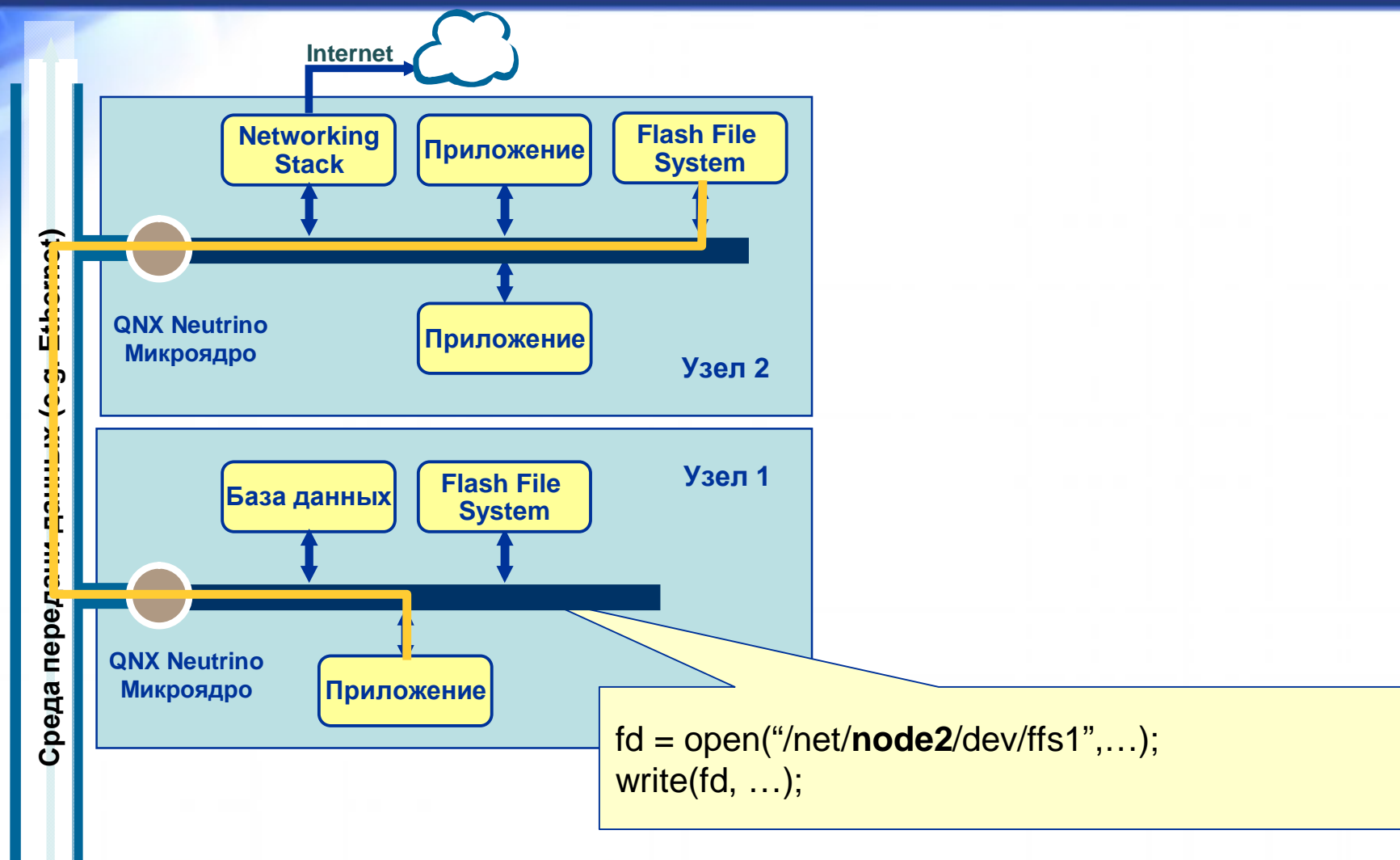
1. Сбой драйвера из-за некорректного обращения с памятью
2. Ядро уведомляет CPM об ошибке
3. Сохраняется отладочная информация о процессе (стандартный core файл)
4. Драйвер выгружается и возвращает ресурсы; уничтожается IPC канал
5. CPM перезапускает новый драйвер
6. Каналы IPC драйвера восстанавливаются клиентской библиотекой CPM
7. Драйвер запрашивает информацию у CPM о своем состоянии в последней контрольной точке и сервис восстанавливается

# Прозрачные сетевые вычисления





# Прозрачные сетевые вычисления



# Прозрачные сетевые вычисления для резервирования

Dual Control Cards



I/O Cards



```
fd = open("/health",...);  
write(fd, ...);
```

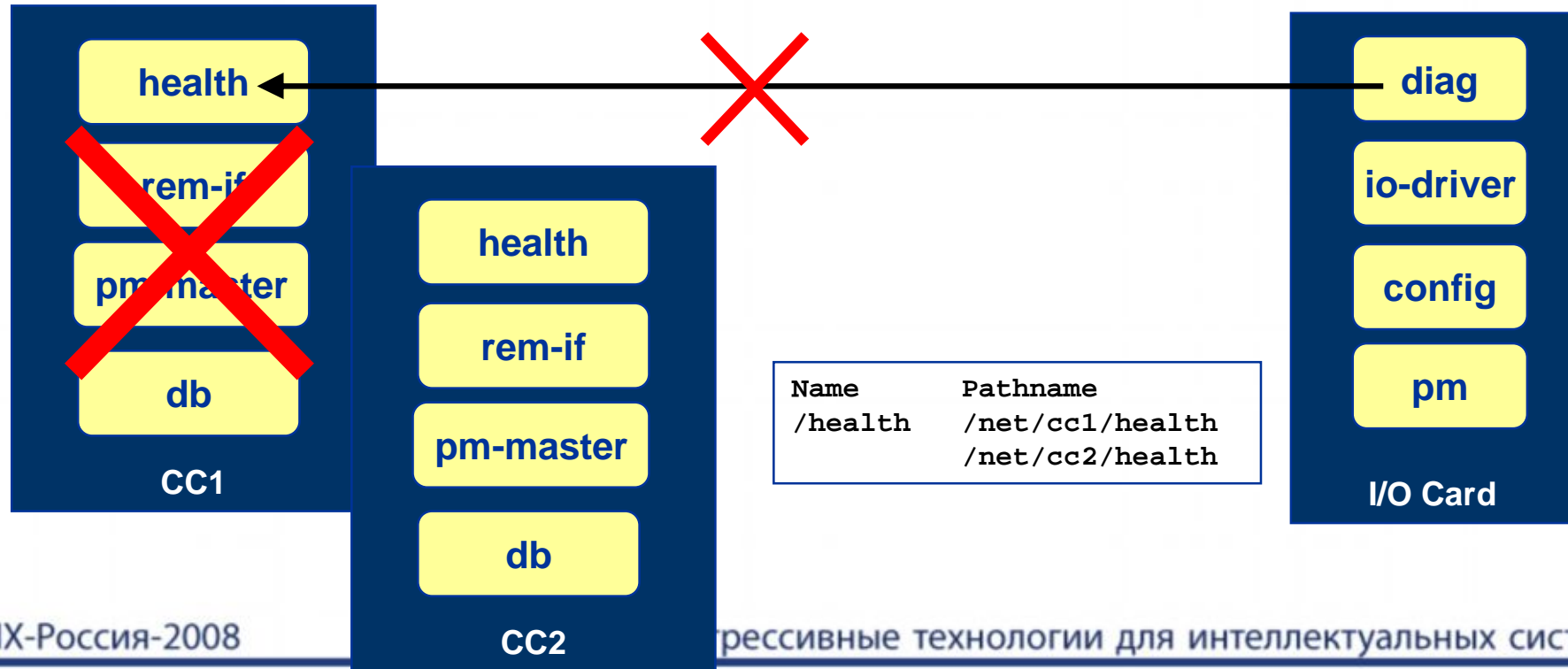


# Прозрачные сетевые вычисления для резервирования

Dual Control Cards



I/O Cards



# Прозрачные сетевые вычисления для резервирования

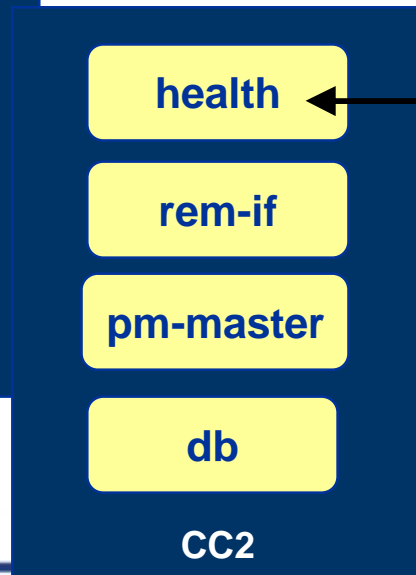
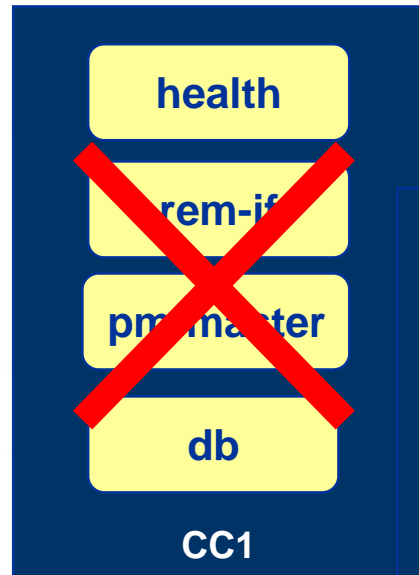
Dual Control Cards



I/O Cards



```
fd = open("/health",...);  
write(fd, ...);
```



Name	Pathname
/health	/net/cc1/health
	/net/cc2/health

# Адаптивная декомпозиция для увеличения готовности

Используйте Адаптивную Декомпозицию для:

- Защиты от подсистем, пытающихся захватить все ресурсы
- Гарантий времени CPU восстановления после сбоев
- Гарантий времени CPU для пользовательских интерфейсов и удаленного управления

Высокая готовность

CPM Guardian

Critical Process Monitor

health

Управление сбоями

Интерфейсы управления



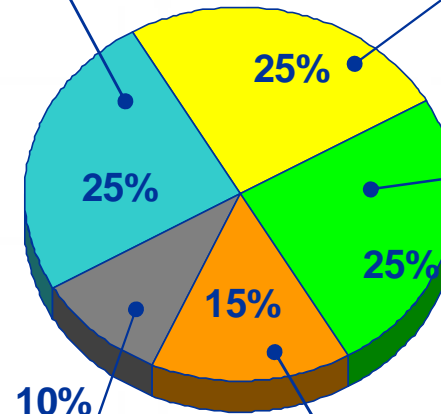
rem-if



pm-master

Мониторинг производительности

db





# Масштабируемость

# Что если система не масштабируется?



→ **Дорого разрабатывать**

- > Много разработчиков
- > Нет повторного использования наработок
- > Нет единого центра компетенции

→ **Долго разрабатывать**

- > Потери времени при смене ОС
- > Каждый раз разработка начинается заново

→ **ОС ограничивает развитие компании!!**

# За счет чего система масштабируется?

## → Масштабирование аппаратной платформы

- > Распределенные многопроцессорные системы для высокопроизводительных комплексов
- > Многопроцессорные системы для нагруженных узлов

## → Масштабирование программного обеспечения

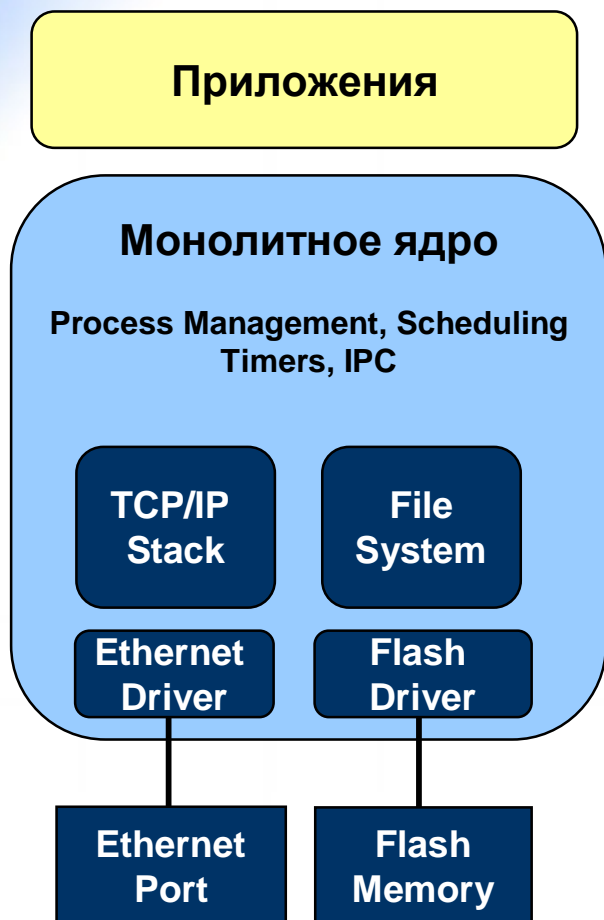
- > Архитектура ПО должна поддерживать распределенные вычисления
- > ПО должно поддерживать многоядерные процессоры и параллельные вычисления

## → Масштабирование групп разработчиков

- > Построение больших масштабируемых систем – это также вопрос человеческого ресурса
- > Инструменты и техники разработки крайне важны!



# Распределенные вычисления с монолитным ядром



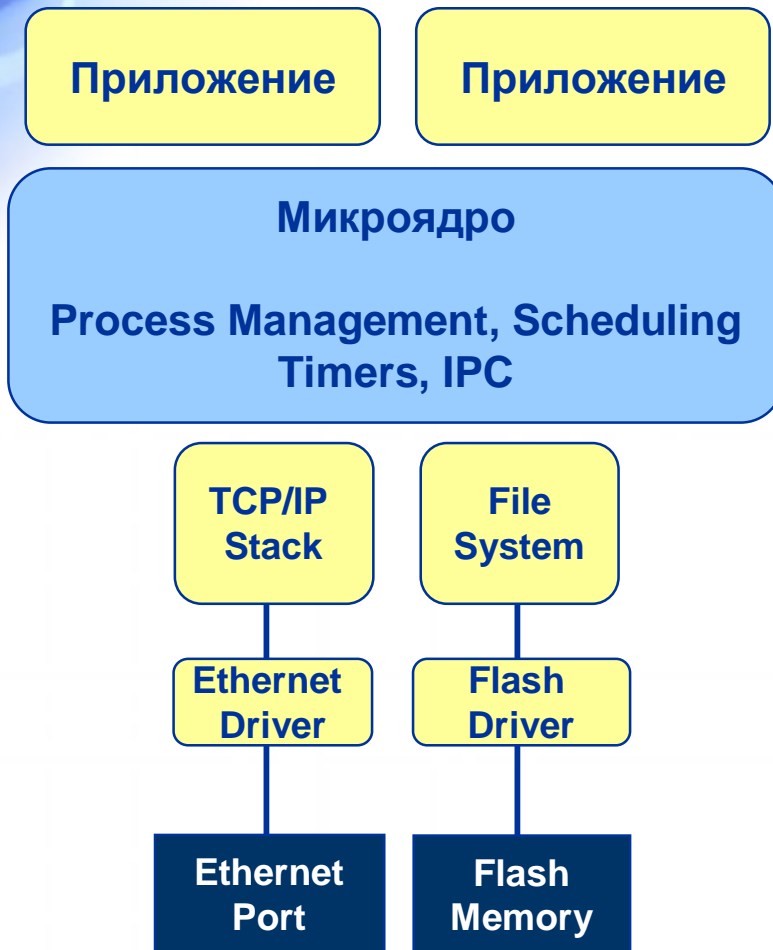
→ В монолитном ядре сервисы ОС встроены в ядро

- > Сетевые стеки
- > Файловые системы
- > Драйвера устройств

→ Распределенные вычисления возможны только на прикладном уровне

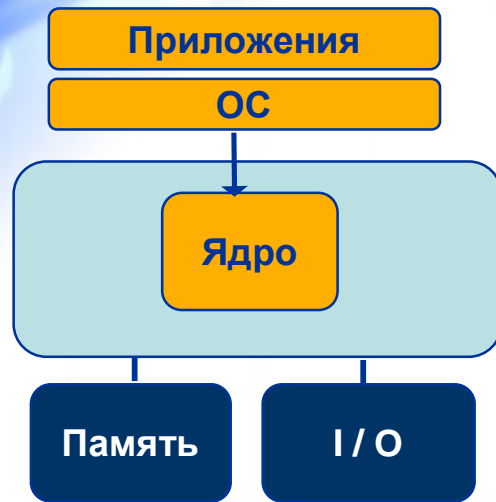
- > TCP/IP стек, файловые системы, драйвера устройств не «адресуются» внешними системами
- > Межпроцессное взаимодействие возможно на прикладном уровне и может быть расширено для использования в сети

# Распределенные вычисления в QNX Neutrino

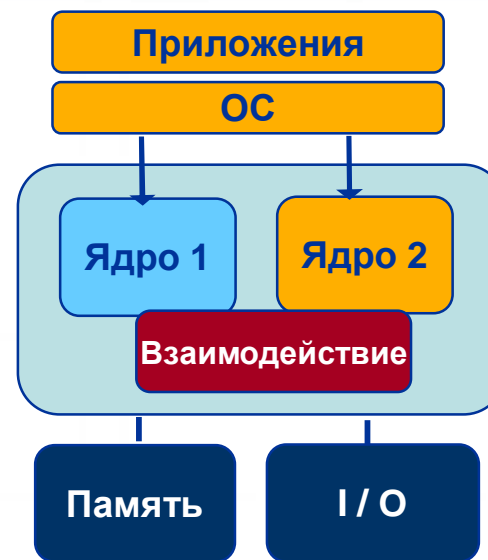


- **Сервисы ОС как обычные приложения вне ядра**
  - > **Сетевой стек**
  - > **Файловые системы**
  - > **Драйвера устройств**
- **Сервисы в виде приложений могут быть использованы как локально, так и удаленно по сети**
  - > **Внешние приложения могут «адресовать» сервисы ОС на узле**
  - > **Защита и контроль доступа осуществляется на уровне процессов**

# Масштабирование производительности



Одно ядро



Мультикор



Дискретная мульти-  
процессорная система

- Добавляются процессоры или ядра процессора
- ПО масштабируется через SMP и BMP
- Прозрачные сетевые вычисления абстрагируются от границ процессоров и узлов

# Масштабирование групп разработчиков - AP



Восстановление

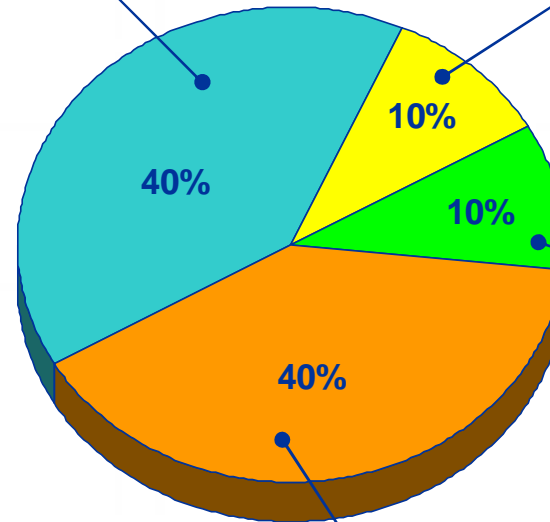


Интерфейсы управления

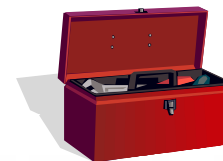


Разработка системы

- Системные архитекторы распределяют ресурсы системы между подсистемами / группами разработчиков
- Группа разработчиков может как угодно настраивать систему внутри своей партии
- Проверка наихудшего сценария загрузки



Управление сбоями



Контроль производительности



# Масштабируйте в QNX мир сетевых решений NetBSD

→ Прол  
люб  
QNX  
на у

## НОВЫЕ стеки протоколов

### Лучший сетевой стек NetBSD в надежных устройствах под управлением ОСРВ QNX

- Актуальные версии протоколов и высокая производительность
- Устойчивость работы и динамическое обновление, без остановки работы системы
- Простота добавления дополнительных модулей NetBSD, за счет совместимости на уровне исходного кода

QNX-Росси

**swd**  
SOFTWARE

196135, Санкт-Петербург, пр. Юрия Гагарина, 23  
115533, Москва, пр. Андропова, 22/30  
info@swd.ru, www.swd.ru

тел: +7 (812) 702-08-33  
тел: +7 (495) 651-61-56



СИСТЕМ

# QNX – единая платформа для всего спектра оборудования



## QNX – единая ОС!!



# Минимизация рисков

# Риски на всех этапах ЖЦ

Планирование

Разработка

Вывод на рынок

Сопровождение



**Технологические партнеры могут снизить часть рисков производителя, а часть рисков взять на себя**



# Технологическое партнерство – путь к минимизации рисков

## → Сокращение управленческих рисков:

- > **Принятие обоснованных решений на этапе планирования**
  - Предсказуемое бюджетирование проекта
  - Точность временных оценок
- > **Партнерские сервисы – средство сокращения ТСО и ТТМ**

## → Сокращение технологических рисков:

- > **Технологический партнер несет ответственность за поставляемые технологии**
- > **Преимущества специализации - каждый занимается тем, в чем он силен: производитель – прикладными задачами, технологический партнер – технологиями**



**SWD Software – технологический партнер  
телекоммуникационных компаний**

## Услуги SWD Software

- **Помощь в решении вопросов программно-аппаратной совместимости**
- **Поставка необходимых инструментальных средств разработки и отладки ПО (включая продукты третьих фирм)**
- **Комплексное обучение специалистов**
- **Совместные работы по портированию и/или разработке сетевых протоколов**
- **Техническая поддержка**
- **Возможность создания стенда на территории компании SWD Software Ltd.**

# Специальное ценовое предложение для телекоммуникационных компаний

- Стоимость модулей среды исполнения различается для разных типов устройств
- Стоимость модулей среды исполнения определяется на основании конечной розничной стоимости изделия на QNX
  - > Для дешевых устройств с конечной стоимостью до 15 000 рублей, ориентировочная стоимость модуля среды исполнения – 8% от конечной стоимости устройства
- Специальное ценовое предложение на модули среды исполнения действительно при тираже от 500 изделий в год.

# SWD Software – технологический партнер

- **OSPV QNX – передовая платформа для производителей сетевого и телекоммуникационного оборудования:**
  - > **Соответствует требованиям отрасли по надежности**
  - > **Масштабируется для устройств любого типа и мощности**
- **SWD предлагает проект, а не технологию:**
  - > **Сокращение управленческих и технологических рисков**
    - **Принятие обоснованных управленческих решений**
    - **Гарантии качества технологий и профессиональный центр компетенции технологического партнера**
  - > **Оптимизация TCO и TTM за счет специализации и разделения задач**

# Вопросы?

Спасибо!

Александр Трофимов  
a.trofimov@swd.ru