



ОАО «Корпорация «Русские системы»

Технологии и инструментальные средства разработки прикладного программного обеспечения критических по безопасности систем

Обеспечение максимального качества и
безопасности СПО при сокращении
затрат на его разработку

Москва, 2010



ОАО «Корпорация «Русские системы»



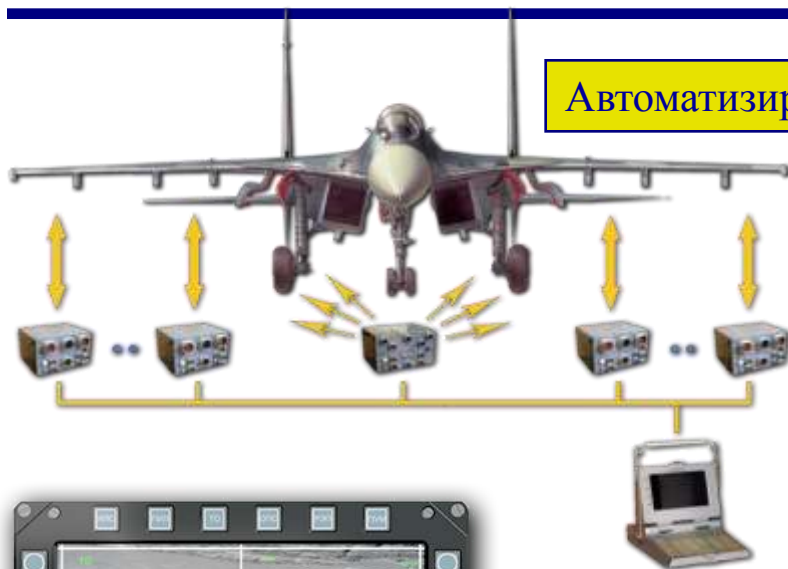
ОАО «Корпорация «Русские системы» основана в 1995 году по инициативе предприятий оборонного комплекса России.

Основные направления деятельности Корпорации – научно-исследовательские и опытно-конструкторские работы, создание высокотехнологичных программно-аппаратных решений на основе мировых достижений в области вычислительной техники, радиоэлектронной базы, информационных технологий и программного обеспечения в интересах Министерства Обороны РФ, предприятий оборонного комплекса России и инозаказчика.



Основные направления разработки

Автоматизированные системы контроля



Бортовые вычислительные комплексы
Бортовые активные системы безопасности полетов



Специализированные системы обучения
Тренажер КРО «Овод-МЭ»

Специализированные тренажеры и
автоматизированные медицинские комплексы



Термины и определения

Критическая по безопасности система

– техническая система, отказ или нештатное функционирование которой может повлечь значительный ущерб.

Критическое по безопасности ПО (safety-critical software)

– программное обеспечение, аномальное поведение которого может вызвать возникновение отказа или нештатного функционирования системы, возможно влекущих значительный ущерб.

не путать с

Критическое по защищенности ПО (security-critical software)

– программное обеспечение, в котором получение несанкционированного доступа к информации (данным или процессам их обработки) может повлечь значительный ущерб.



Роль программного обеспечения (ПО) в вопросах безопасности

На практике в современных программно-аппаратных комплексах до
85-90% отказных ситуаций связано с ошибками в ПО

С учетом тенденции по расширению объемов задач, решаемых в ПО,
вклад ПО в надежность и безопасность систем постоянно возрастает

Президентом РФ 21 мая 2006 г. утвержден перечень, в соответствии с которым
технологии производства ПО отнесены к критическим технологиям РФ



Как обеспечить заданный уровень безопасности?

Необходимо планировать и осуществлять мероприятия по обеспечению безопасности ПО на протяжении всего жизненного цикла ПО

Def Stan 00-55 Requirements for Safety Related Software in Defense Equipment
DO-178B Software considerations in Airborne Systems and Equipment Certification
IEC 60880(2006) Nuclear power plants – Instrumentation and control systems important to safety
– Software aspects for computer-based systems performing category A functions

...

ГОСТ Р 51904-2002. Программное обеспечение встроенных систем.
Общие требования к разработке и документированию
МАК КТ-178В Требования к программному обеспечению бортовой аппаратуры
и систем при сертификации авиационной техники
ГОСТ РВ 0019-001 Программное обеспечение встроенных систем.
Требования к содержанию и оформлению документов

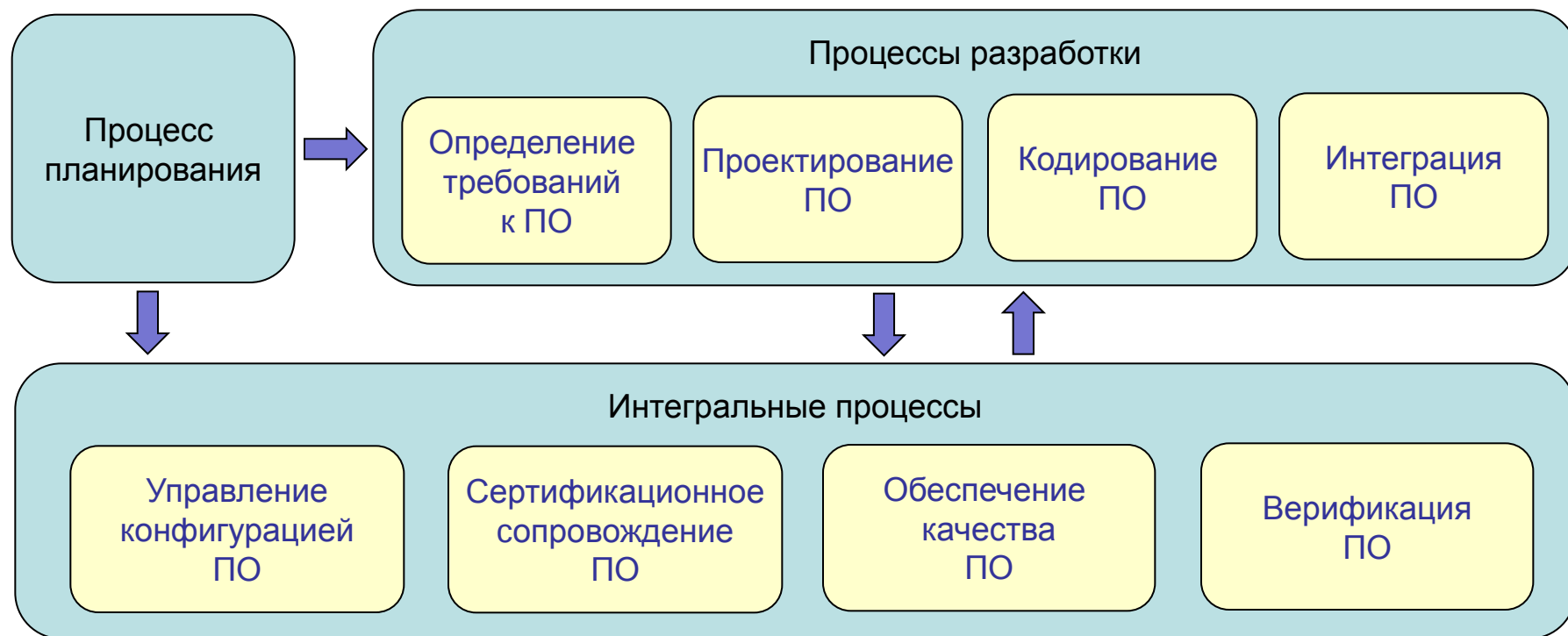
...

Стандартами и другими нормативными документами определяются процессы разработки программного обеспечения и устанавливаются цели, которые необходимо достичь в ходе выполнения соответствующих процессов.



Типовые процессы жизненного цикла ПО

ГОСТ Р 51904



Категории отказных ситуаций и уровни ПО

ГОСТ Р 51904

Категория А: Катастрофическая

Препятствует безопасному функционированию объекта управления

Категория В: Опасная/Критическая

Приводит к критическому уменьшению возможностей объекта управления или способности персонала справиться с неблагоприятными режимами

Категория С: Существенная

Приводит к существенному снижению возможностей объекта управления или способности персонала справиться с неблагоприятными режимами

Категория D: Несущественная

Незначительно уменьшает безопасность объекта и требует действий персонала, которые осуществимы в пределах их возможностей

Категория Е: Невлияющая

Не воздействует на эксплуатационные возможности объекта управления и не увеличивает рабочую нагрузку персонала

Необходимо определить уровень ПО исходя из того, какой категории отказную ситуацию может создать его аномальное поведение



Цели, достигаемые в процессе разработки ПО

ГОСТ Р 51904

Например, «Разработать архитектуру ПО», «Убедиться, что алгоритмы точны и корректны», «Убедиться, что исходный код согласуется с архитектурой ПО» и т.п.

Уровень ПО (категория отказной ситуации)	Целей, удовлетворяемых обязательно с обеспечением независимости	Целей, удовлетворяемых обязательно	Целей, удовлетворяемых на усмотрение заказчика
A: Катастрофическая	25	41	1
B: Опасная	14	51	2
C: Существенная	2	55	10
D: Несущественная	2	28	37
E: Невлияющая			67

Всего 67 целей



1. План сертификации в части ПО
2. План разработки ПО
- ...
16. Описание проекта ПО
- ...
19. Исходный код ПО
20. Исполняемый объектный код ПО
21. Процедуры верификации ПО
- ...
38. Руководство пользователя ПО
39. Описание версии ПО



Всего 39 видов документов

Документы должны содержать, в том числе, результаты работ по достижению поставленных целей, следовательно их объем существенно зависит от уровня ПО

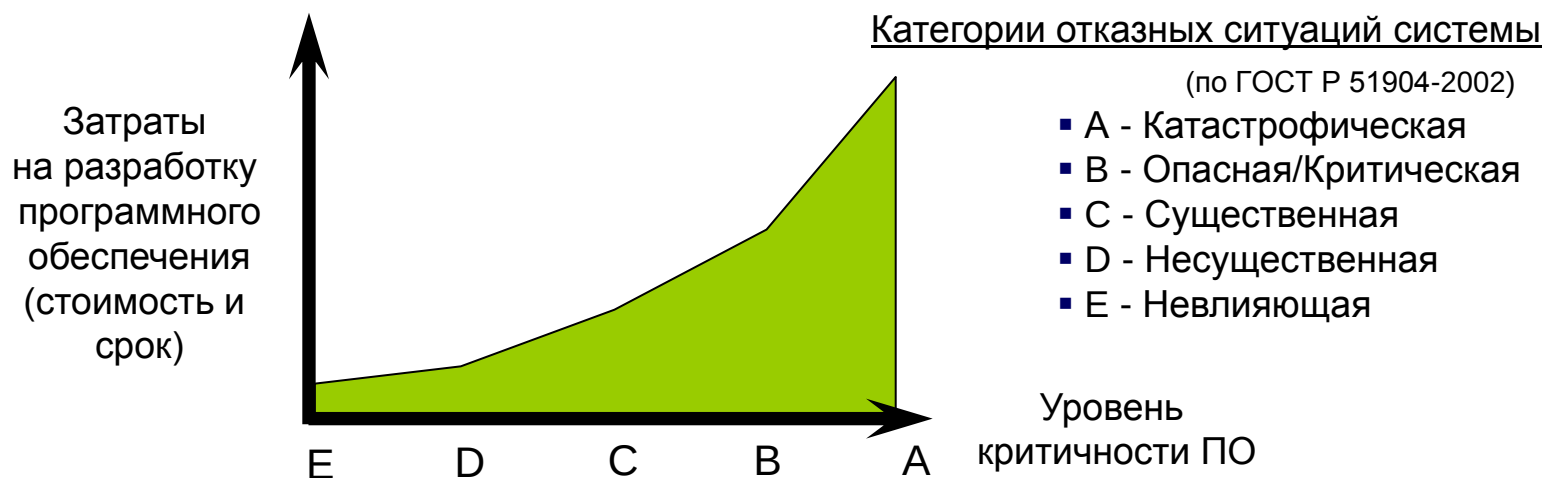


Затраты на разработку специального ПО

В соответствии с нормативными документами:

ГОСТ Р 51904-2002, ГОСТ РВ 0019-001-2006, МАК КТ-178В и др.

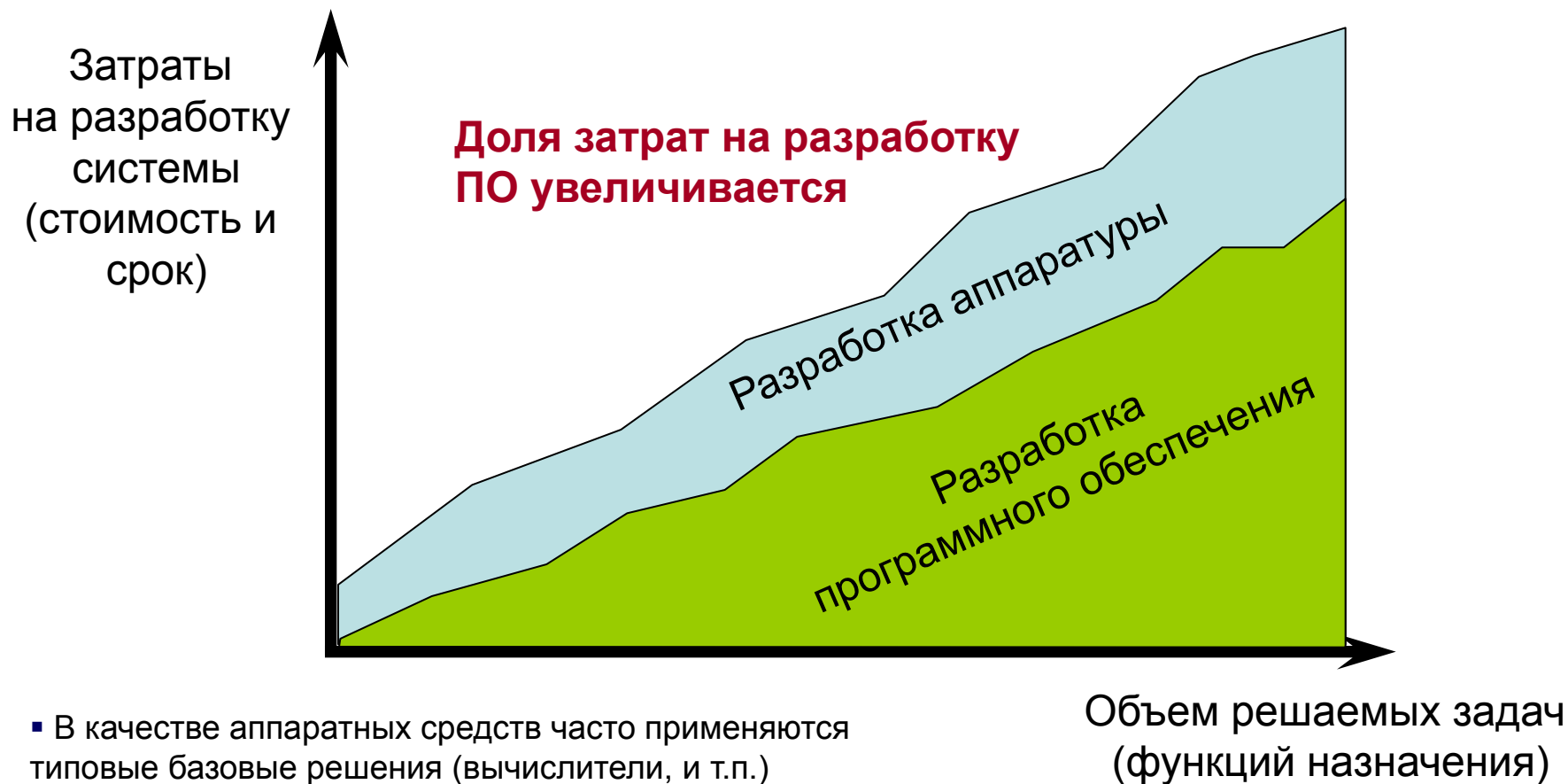
объем работ, выполняемых при разработке специального ПО, зависит от уровня ПО, определяемого серьезностью влияния отказной ситуации на объект управления.



С увеличением уровня критичности программного обеспечения
затраты на разработку растут экспоненциально



Тенденция в структуре затрат на разработку системы



- В качестве аппаратных средств часто применяются типовые базовые решения (вычислители, и т.п.)
- Специализированное программное обеспечения обычно разрабатывается «под систему» с «0»



Комплексная технология разработки специального ПО

В ОАО «Корпорация «Русские системы» разработаны и внедрены **элементы комплексной технологии разработки специального ПО**, обеспечивающие



Многokратное снижение затрат
(стоимости и сроков) на разработку
программного обеспечения



Повышение уровня безопасности
программного обеспечения (снижение
вероятности наличия ошибок)



Элементы комплексной технологии разработки ПО

- Типовые **архитектурные подходы** к построению прикладного программного обеспечения критических по безопасности систем
- Инструментальные **средства автоматизированного проектирования (САПР)** прикладного программного обеспечения
- Инструментальные **средства для анализа и поддержки требований и верификации ПО** на основе формализованных требований
- Автоматизированная **система управления процессами (АСУП) жизненного цикла** программного обеспечения
- Методические материалы по применению комплексной технологии





Типовые **архитектурные подходы** к построению прикладного программного обеспечения критических по безопасности систем



Типовые архитектурные подходы к построению ПО

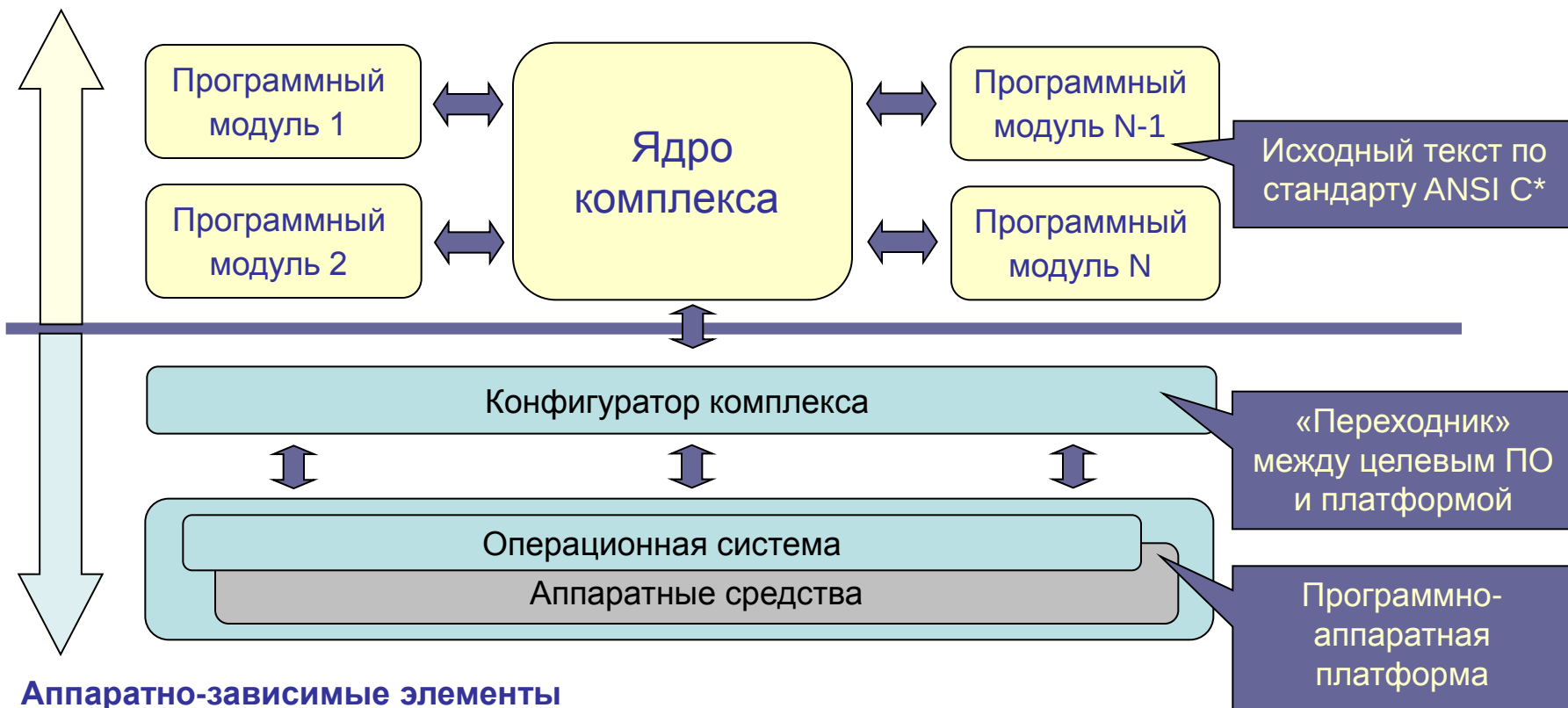
- **Открытая архитектура** – допускает эффективное наращивание функционала, гибкую компоновку системы под конкретное назначение
- **Модульная архитектура** – обеспечивает обособление функций, повторное использования сертифицированных программных модулей
- **Контроль потоков данных и управления** – обеспечивает планирование ресурсов, в том числе режим «жесткого» реального времени
- **Переносимость** – обеспечивает возможность переноса и исполнения ПО без изменения текста на различных ЭВМ и операционных системах
- **Гибкость конфигурирования** – обеспечивает возможность изменения параметров функционирования без изменения текста программы



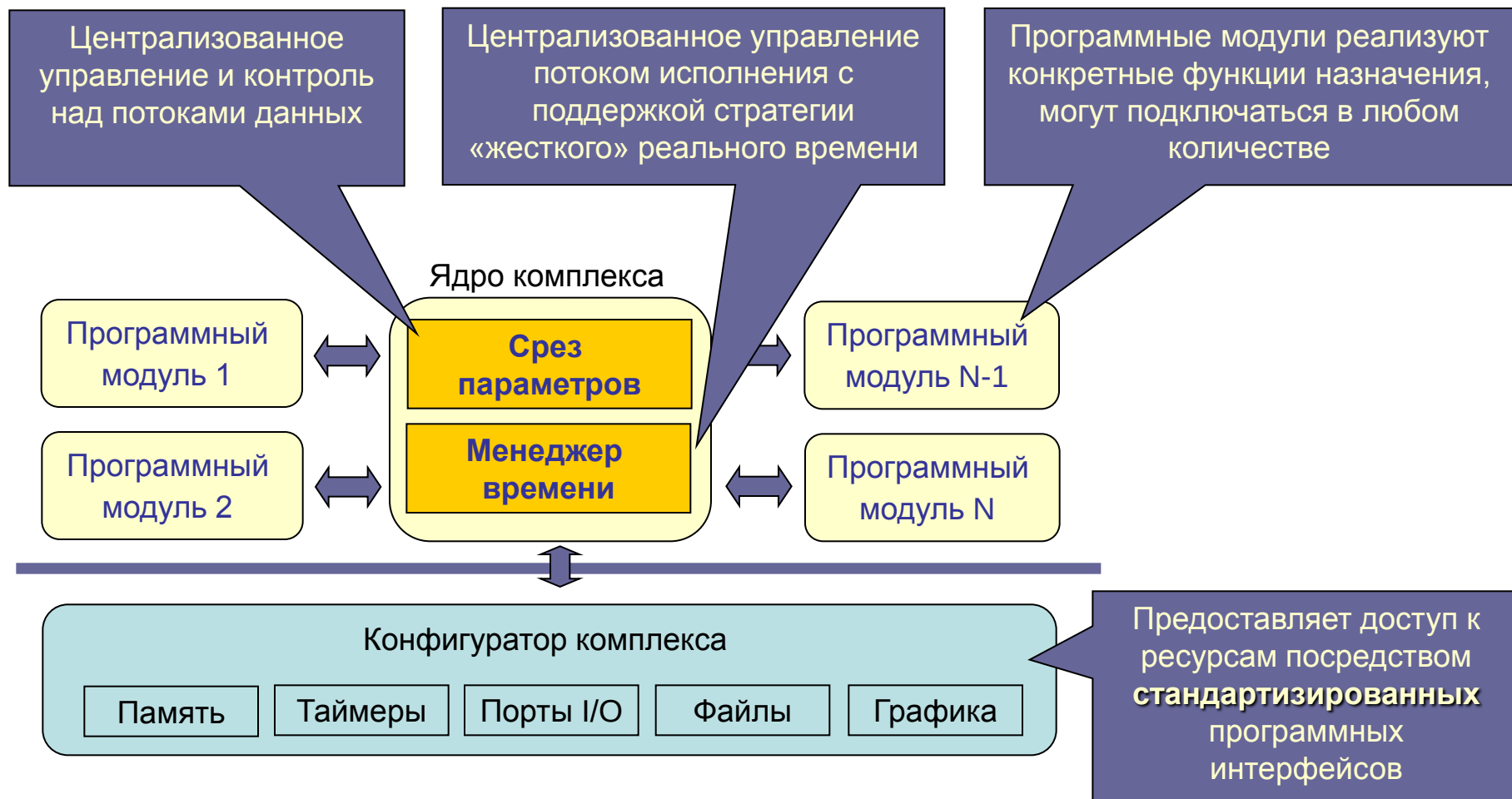
Открытый кросс-платформенный программный комплекс

Шифр «РСДА»

Независимые от платформы программные компоненты



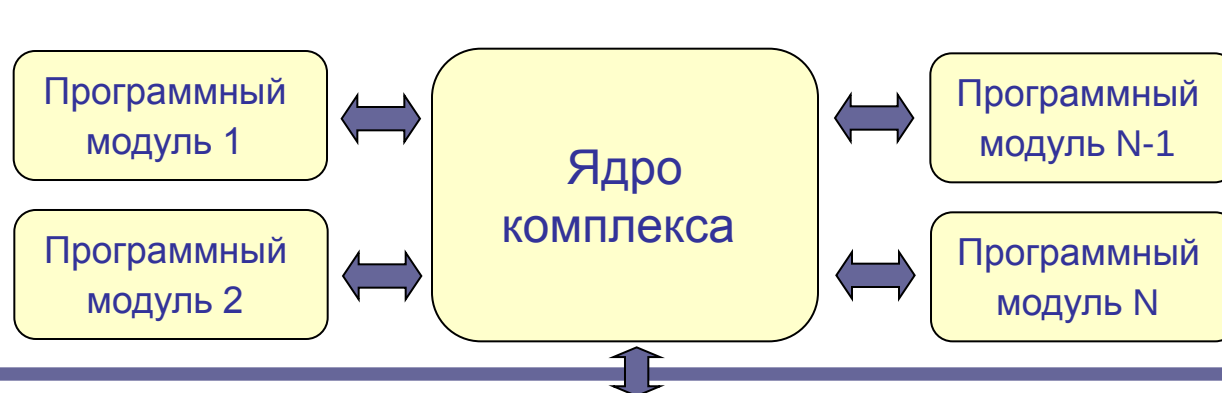
Открытость / Модульность



Кросс-платформенность (переносимость)*

* Независимость от программно-аппаратной платформы

Независимые от платформы
программные компоненты



Intel / Windows

Багет / MCBC

Багет / ОС РВ



Intel / Linux

Пульт АКПА

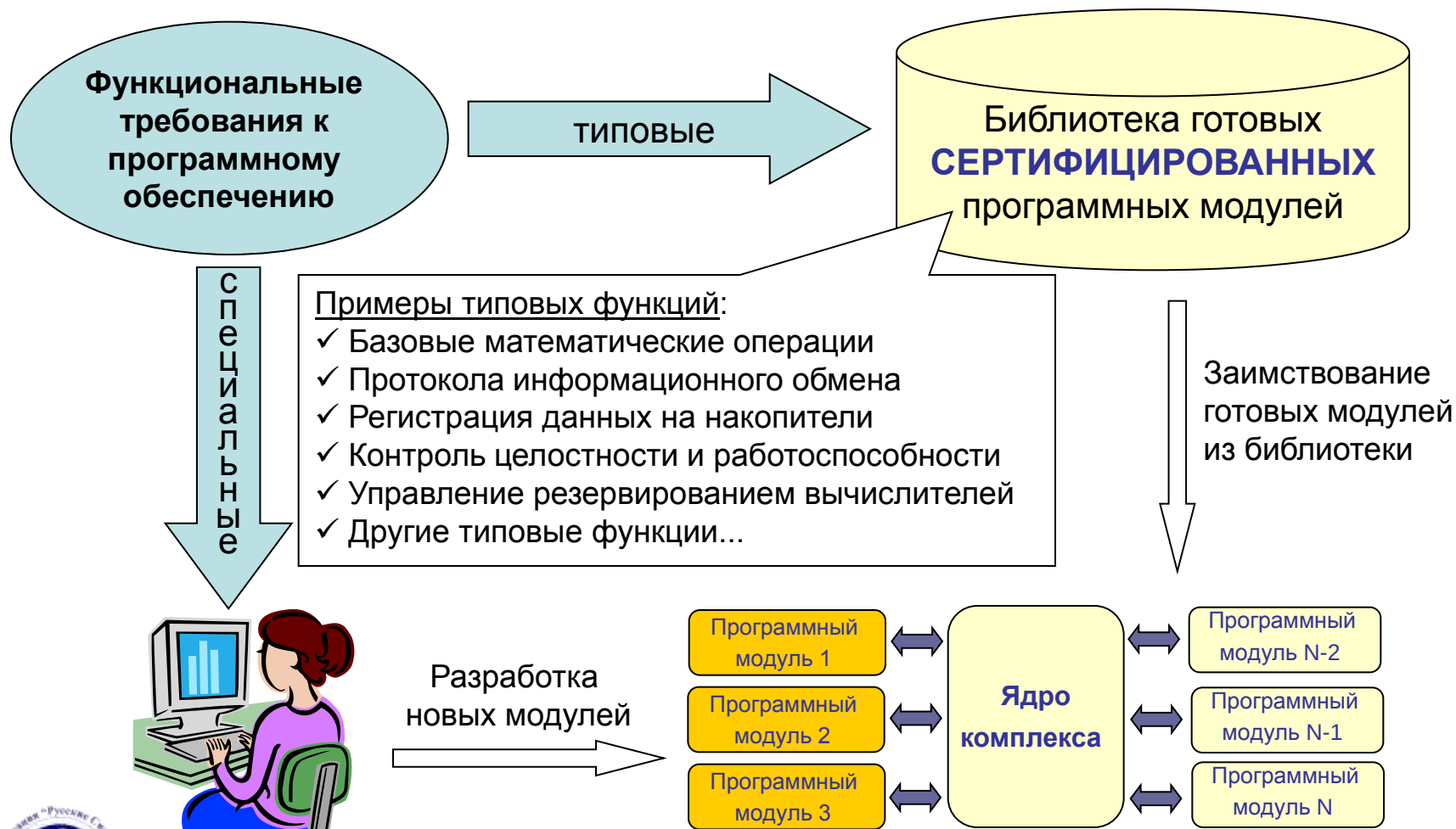
Спец/система

Любая 32-битная платформа с компилятором ANSI C



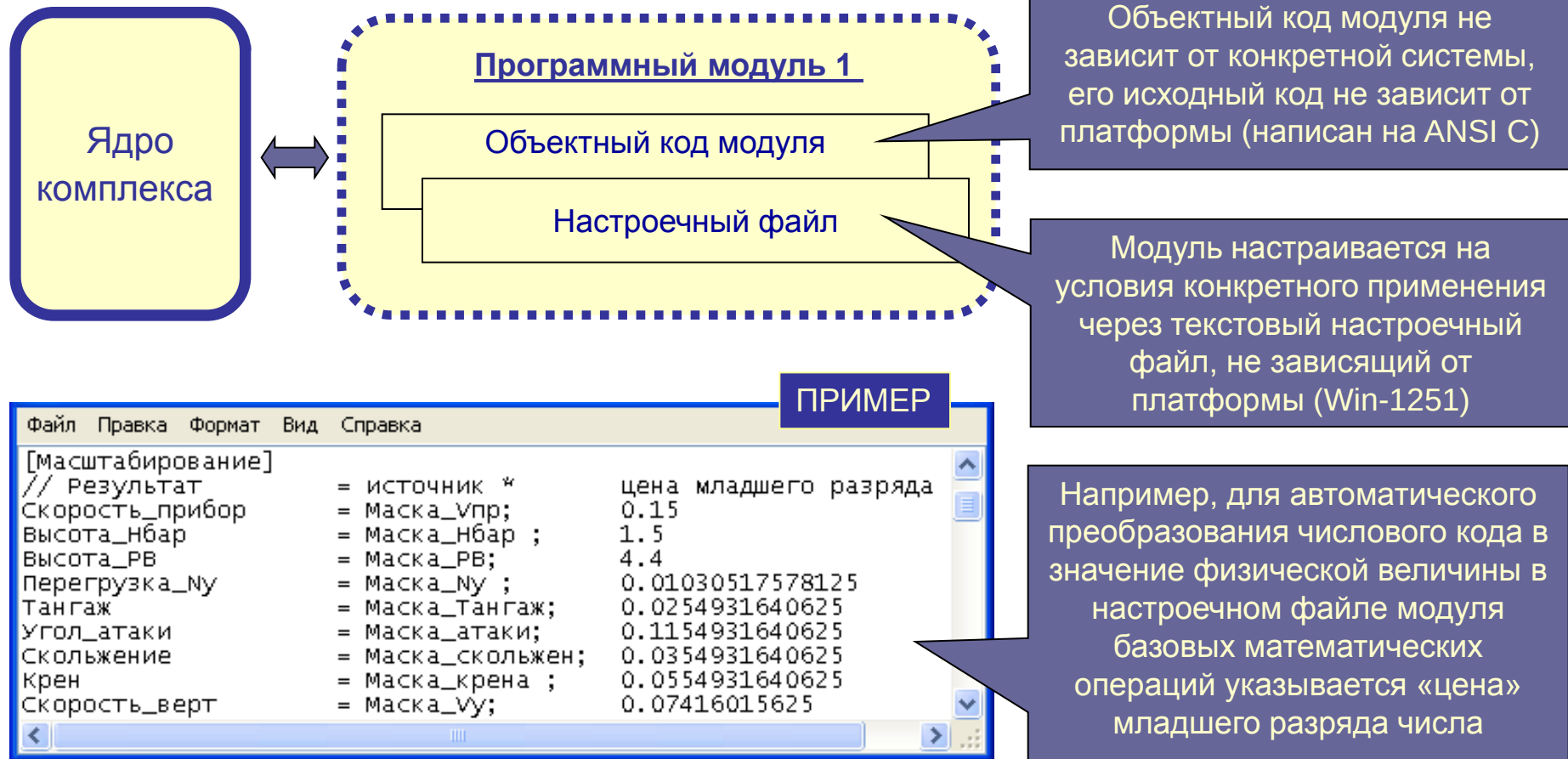
Гибкость конфигурирования (1)

Для типовых функциональных требований создаваемой системы
имеются готовые сертифицированные программные модули



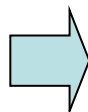
Гибкость конфигурирования (2)

Настройка программного модуля на условия конкретного применения осуществляется специалистом (инженером) через текстовый настроечный файл



Сокращение затрат при разработке ПО

- Открытая архитектура
- Модульная архитектура
- Контроль потоков данных и управления
- Переносимость
- Гибкость конфигурирования

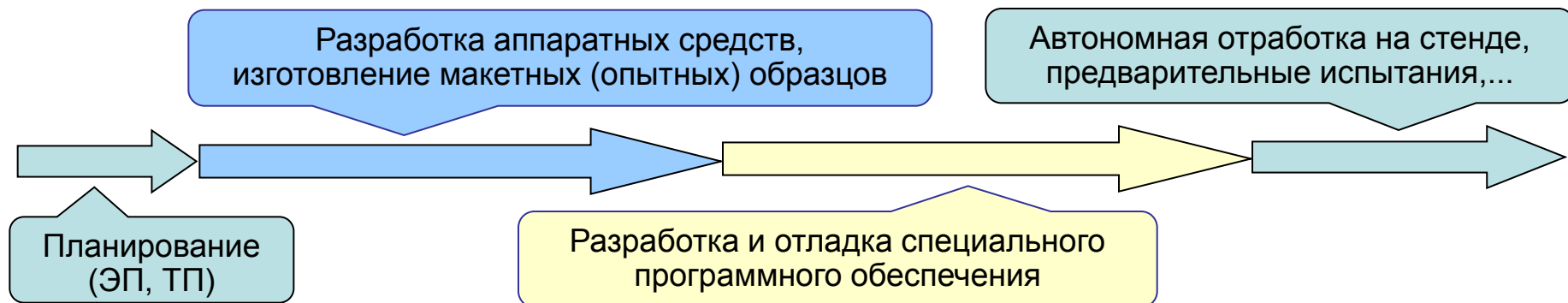


Многokратное снижение затрат
(стоимости и сроков) на разработку
программного обеспечения

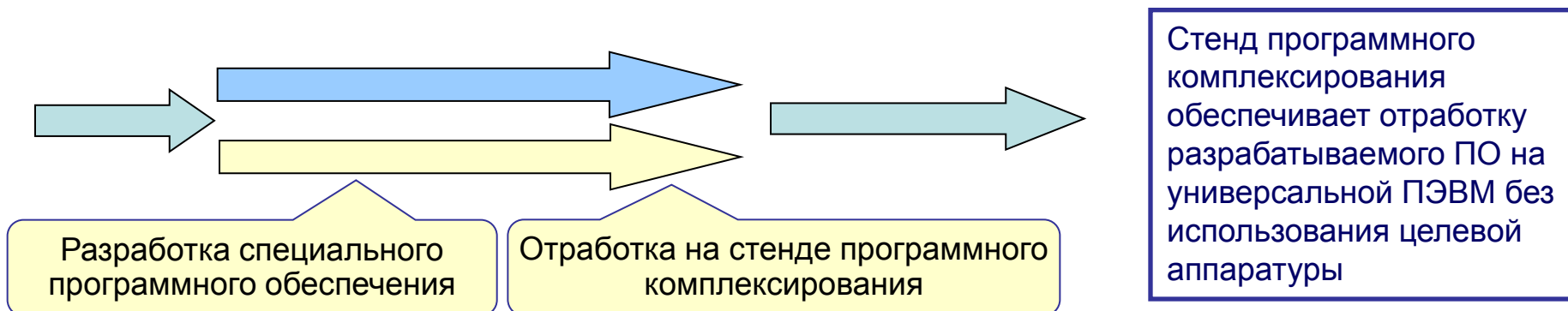


Разработка ПО без целевой аппаратуры

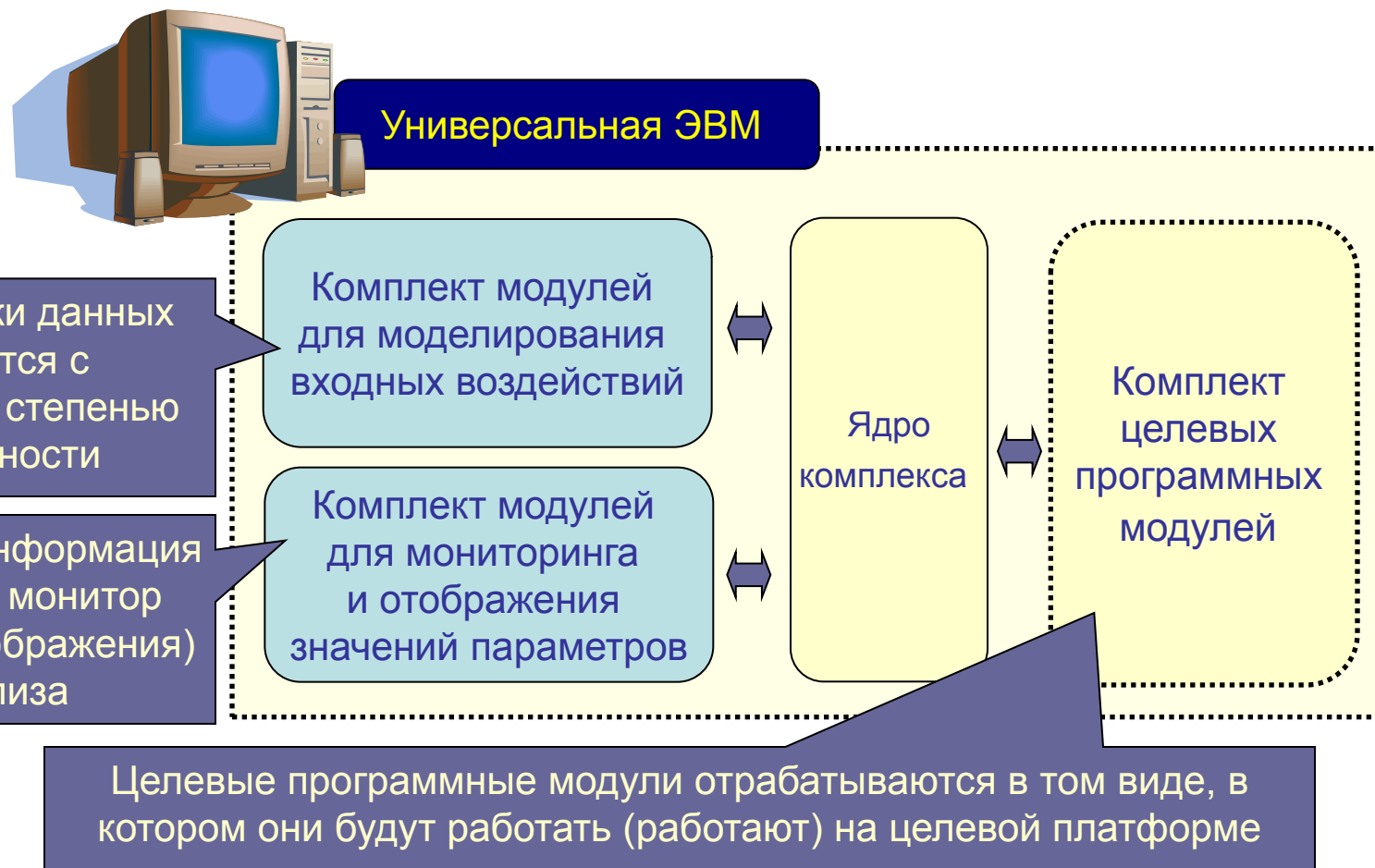
«Классический» цикл разработки



Цикл разработки на базе комплекса РСДА



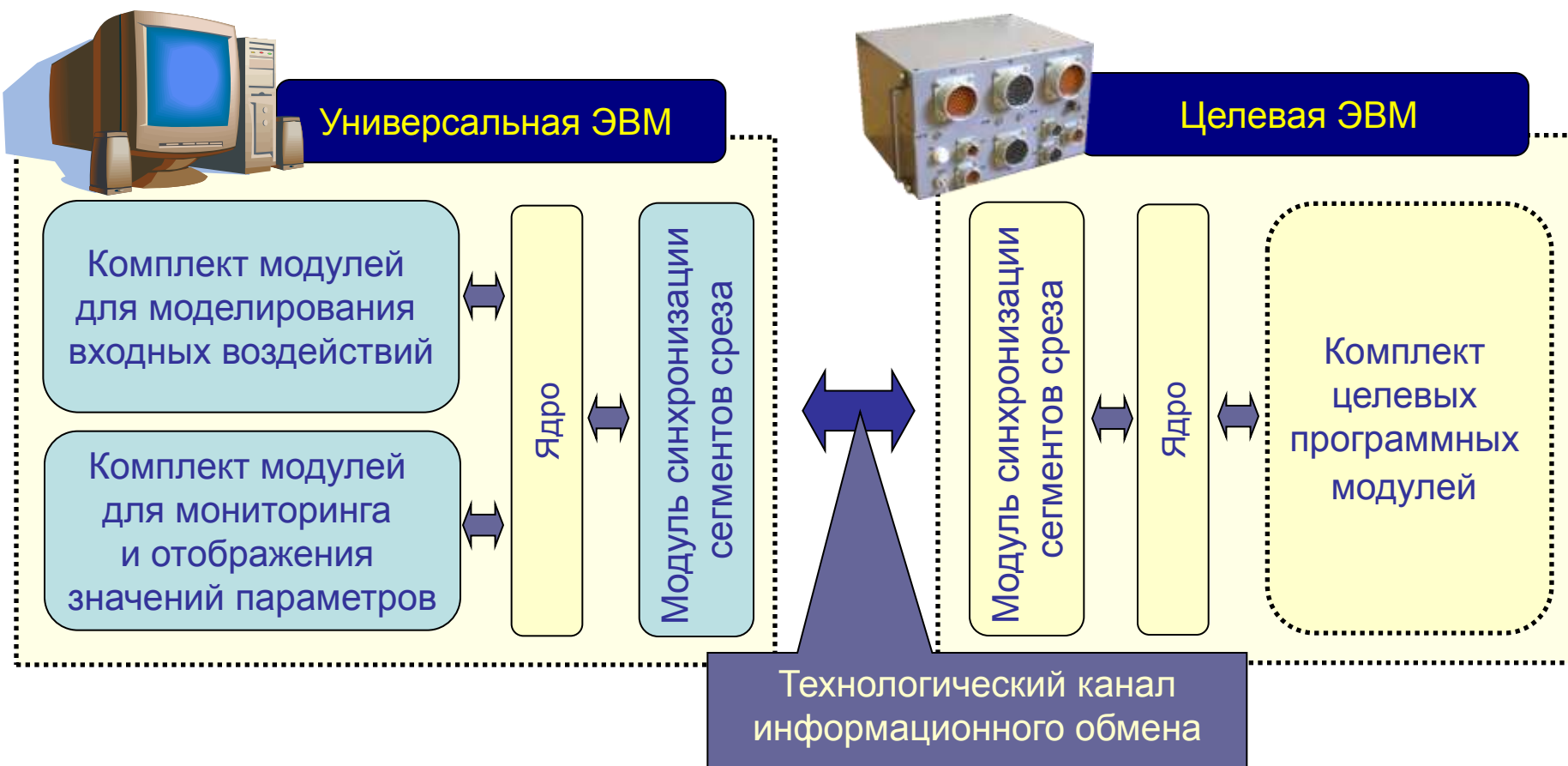
Стенд программного комплексирования



На универсальной ЭВМ имитируется штатное функционирование программного обеспечения целевого комплекса с гарантированной 100% достоверностью



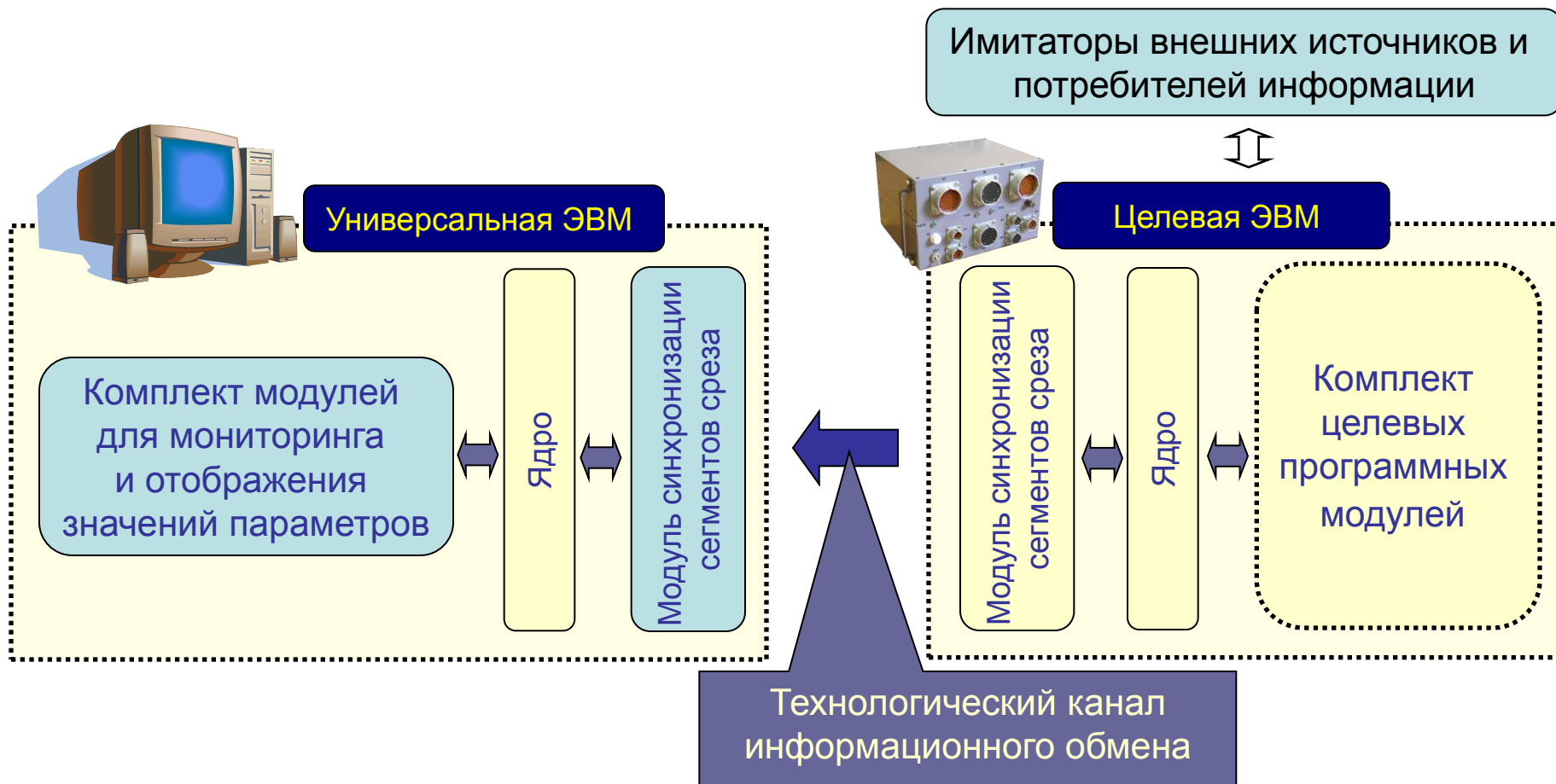
Стенд программно-аппаратного комплексирования



С помощью технологического канала информационного обмена обеспечивается имитация работы программного обеспечения в штатных условиях на целевой вычислительной платформе с гарантированной 100% достоверностью



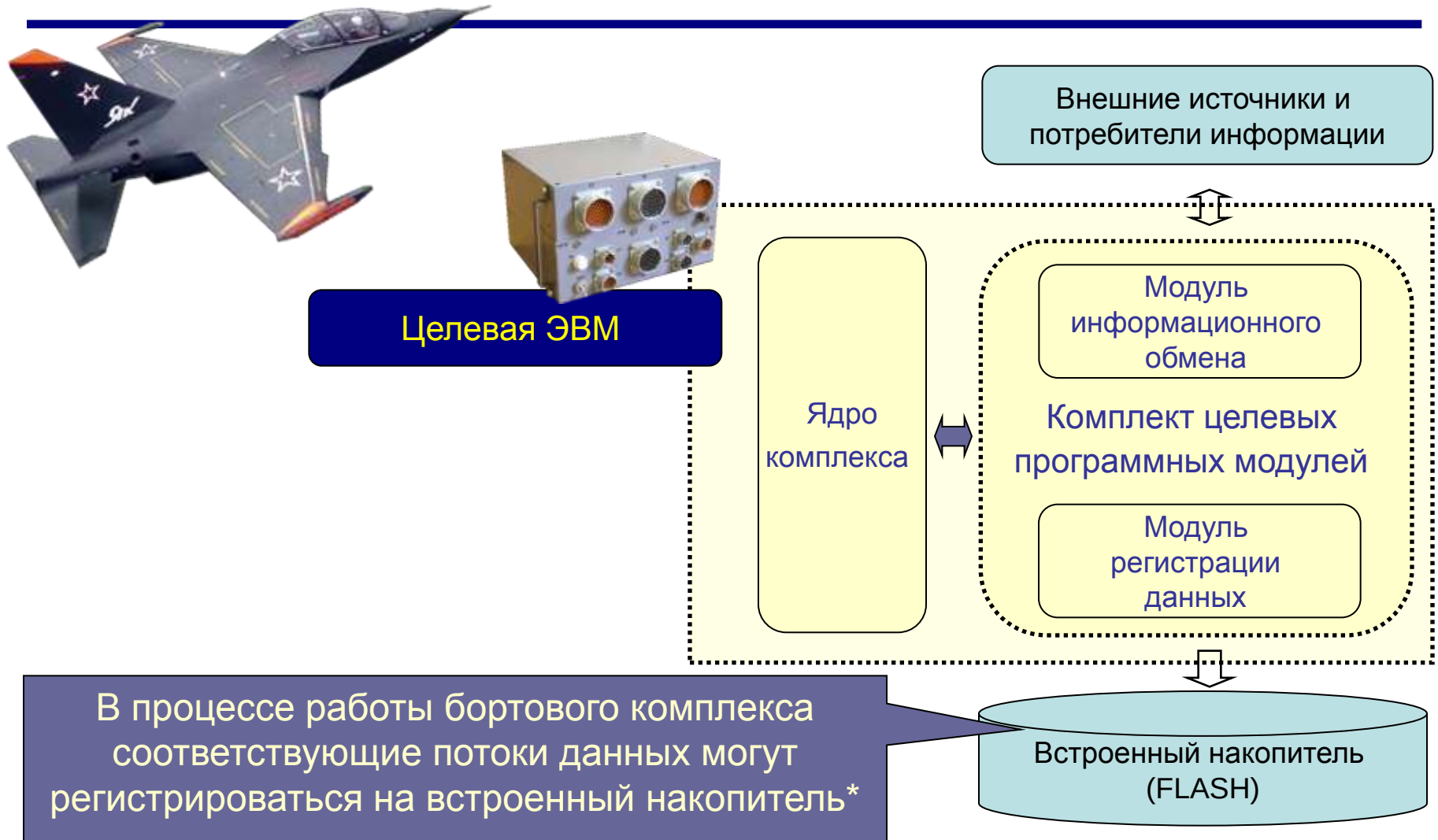
Стенд полунатурного моделирования



На данном стенде осуществляется мониторинг работы комплекса в полунатурных условиях с целью проверки работы штатных средств информационного обмена, функционирование программного обеспечения проверено на предыдущих этапах



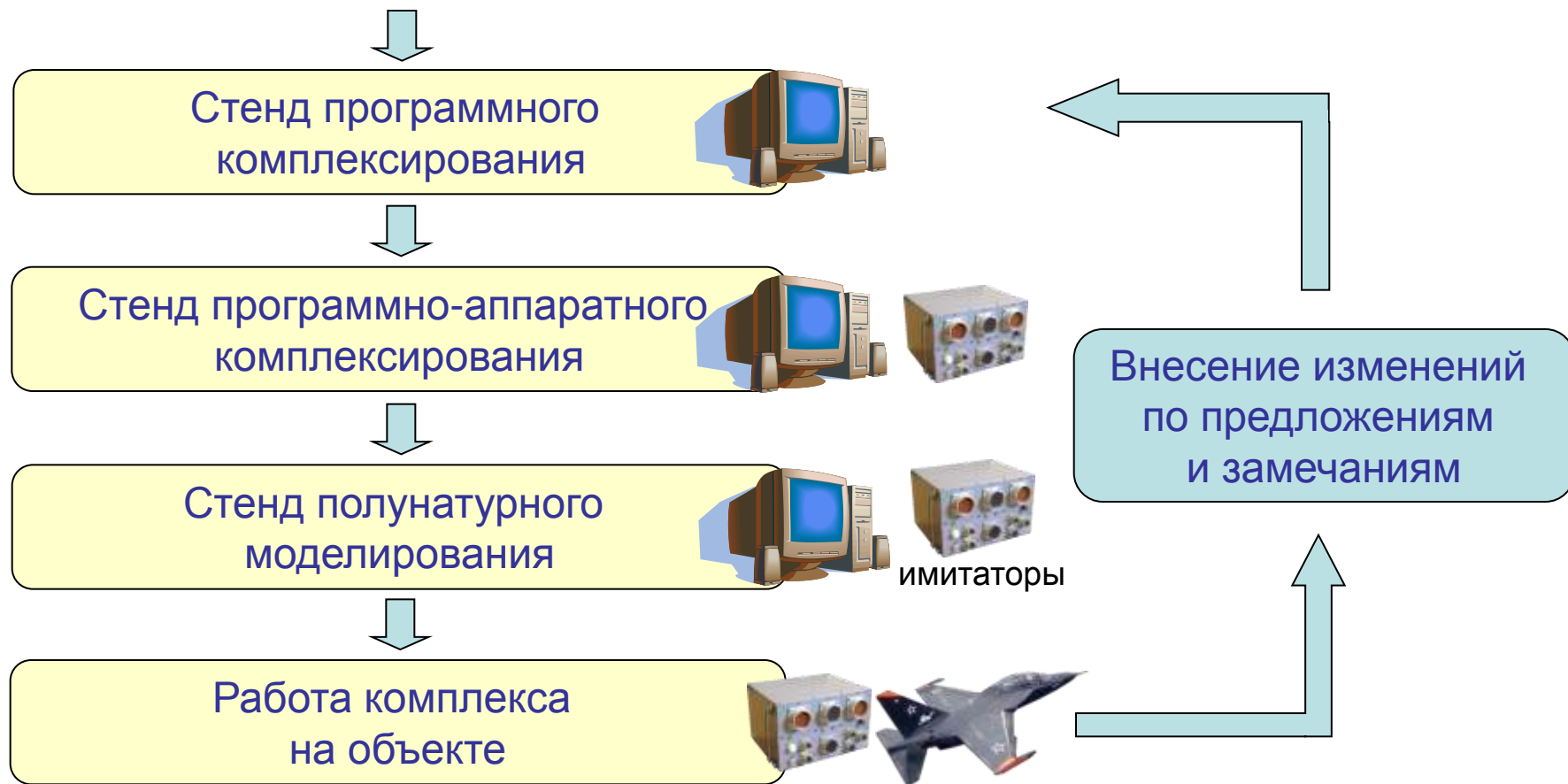
Работа комплекса на объекте



* при наличии эксплуатационного накопителя в составе комплекса



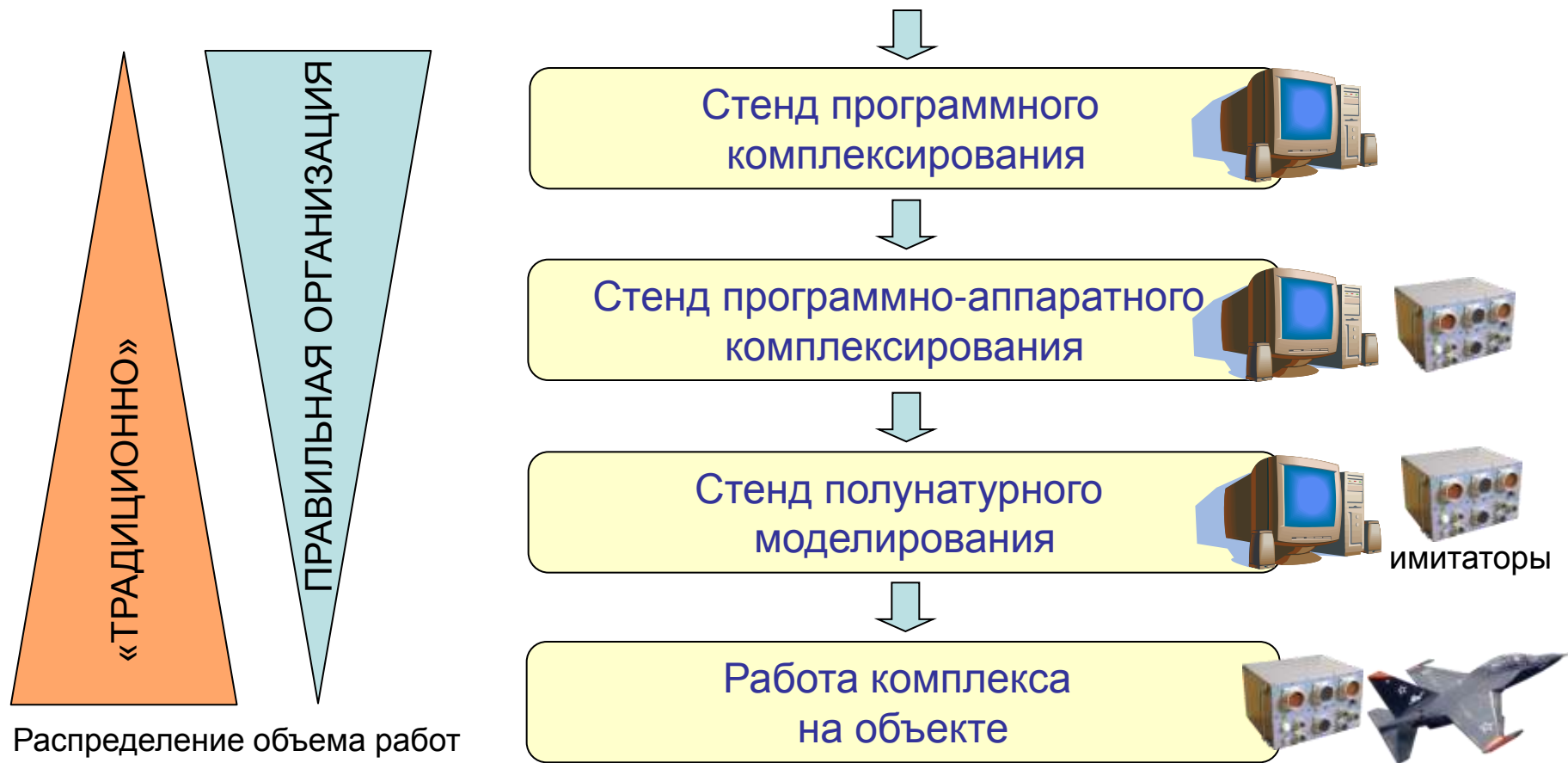
Цикл отработки программного обеспечения



После внесения изменений в программное обеспечение по замечаниям и предложениям, полученным в процессе эксплуатации, осуществляется отработка на стендах различных уровней, в объеме, зависящем от существа изменений в ПО



Распределение объемов работ в цикле отработки ПО

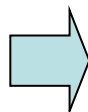


**Чем «ближе» к объекту ведутся работы, тем они дороже,
причем стоимость растет многократно**



Повышение уровня безопасности при разработке ПО

- **Открытая архитектура**
- **Модульная архитектура**
- **Контроль потоков данных и управления**
- **Переносимость**
- **Гибкость конфигурирования**



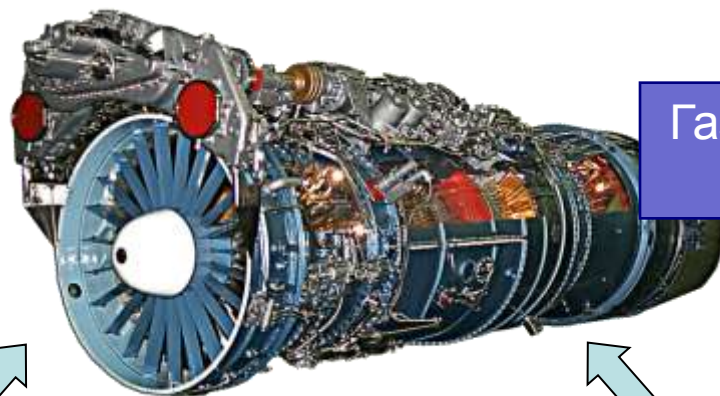
Повышение уровня безопасности
программного обеспечения (снижение
вероятности наличия ошибок)



Многоверсионное неидентичное ПО (1)



Пример



Газотурбинный
двигатель

Электронная
система
управления



Резервная
гидро-механическая
система

Требования по выдерживанию режимов работы ГТД

Надежность повышается за счет использования
систем, построенных на разных физических принципах



Многоверсионное неидентичное ПО (2)

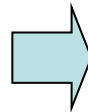


*) в том числе, «неверная» реализация функций в случае отсутствующих (неполных) требований



Гибкое управление вычислительными ресурсами

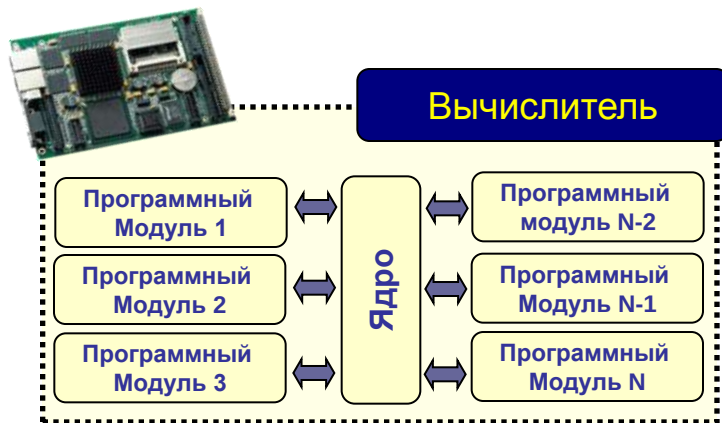
- **Открытая архитектура**
- **Модульная архитектура**
- **Контроль потоков данных и управления**
- **Переносимость**
- **Гибкость конфигурирования**



Гибкое управление ресурсами
вычислительной системы при
организации выполнения ПО



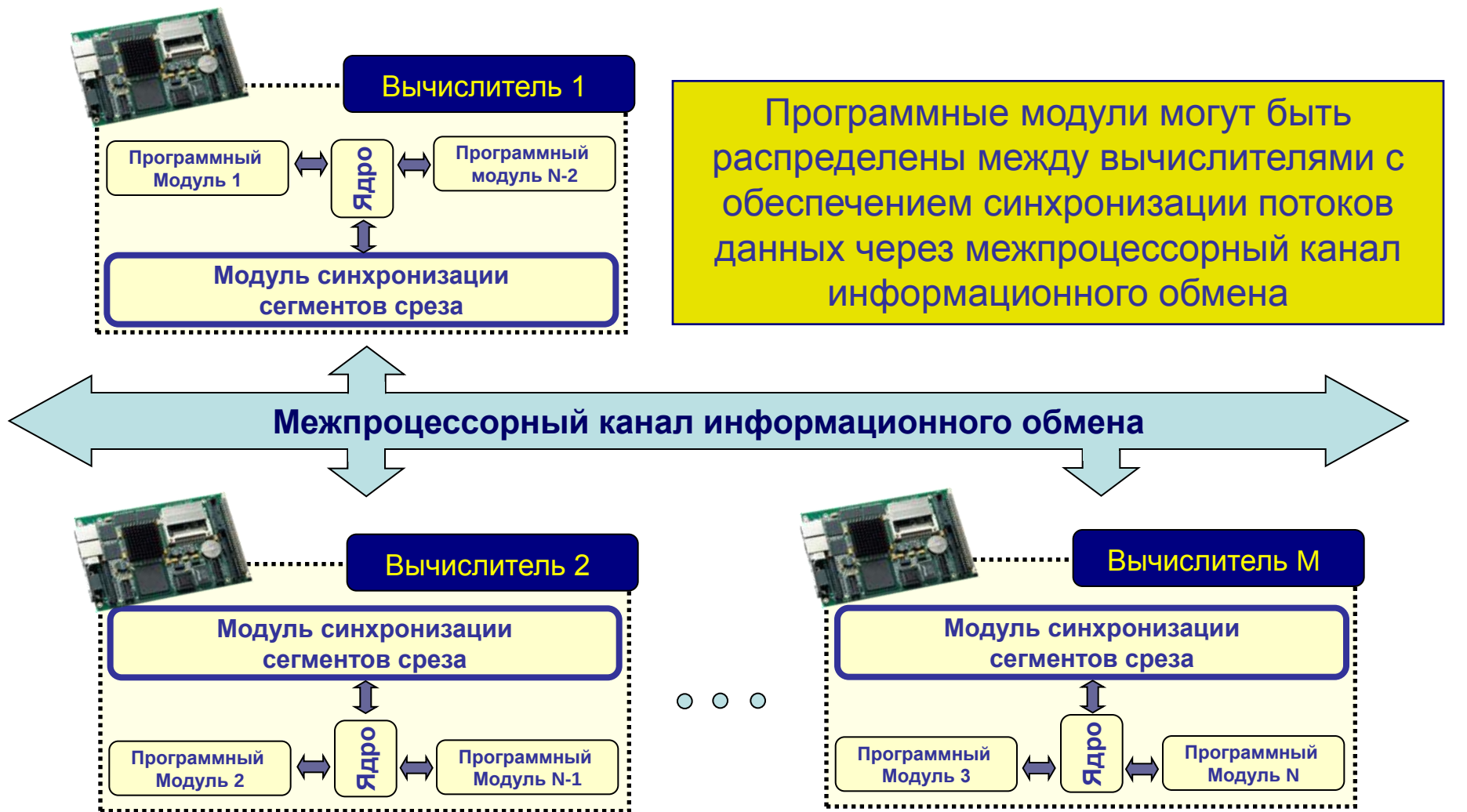
ПО многопроцессорной системы (1)



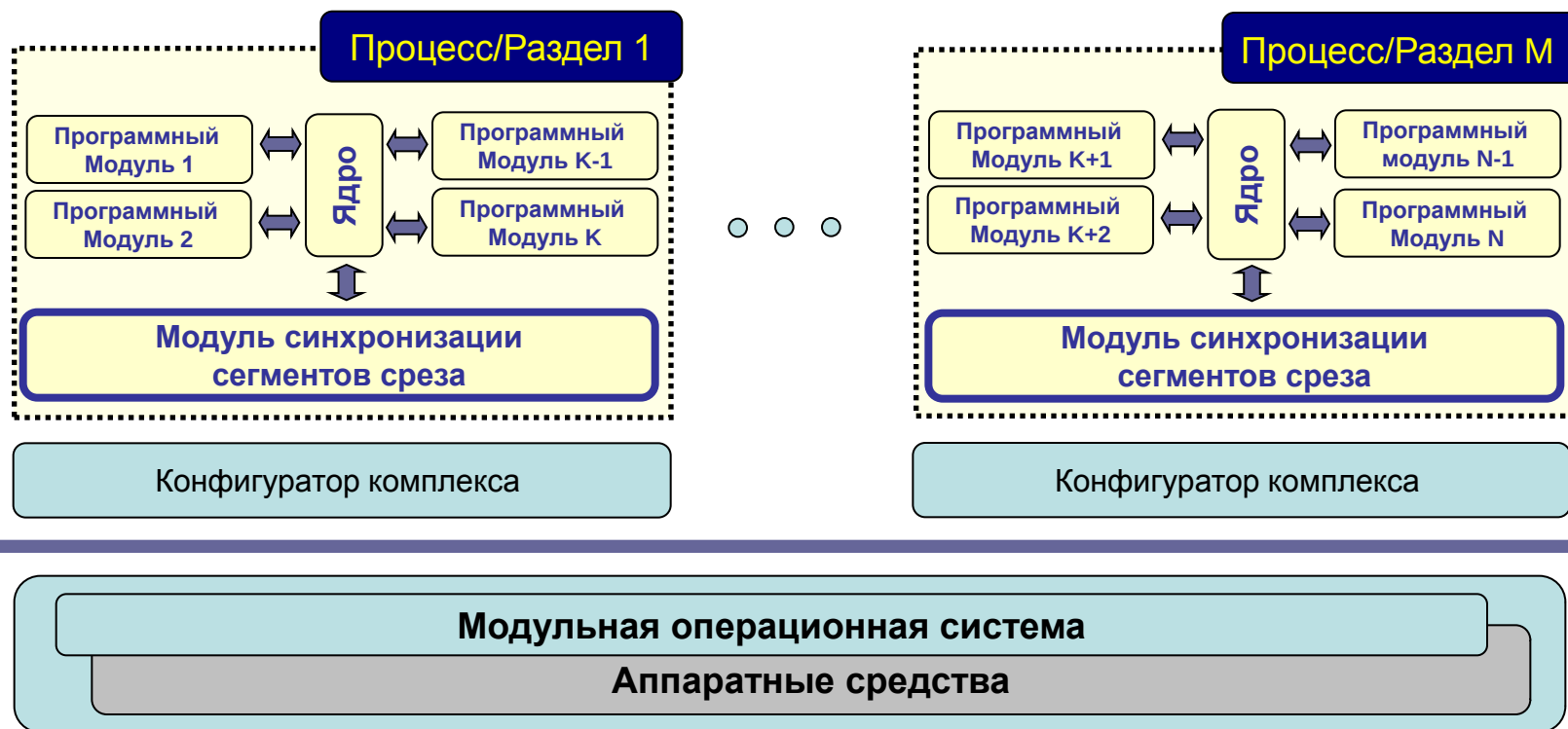
**Не хватает
вычислительных ресурсов?
Требуется
резервирование вычислений?**



ПО многопроцессорной системы (2)



Исполнение ПО в нескольких процессах



Несколько программных комплексов могут работать на одном вычислителе в нескольких процессах или изолированных разделах, в том числе, в соответствии с ARINC-653





Инструментальные средства автоматизированного проектирования (САПР) прикладного программного обеспечения



Разработка алгоритмического обеспечения (1)

«Классический» цикл разработки

Оперирует понятиями:
сигнал, циклограмма,...

1

Инженер*
разрабатывает алгоритм
работы системы

Задание,
исходные данные

Оперирует понятиями:
переменная, функция,...

2

Программист
реализует алгоритм
в ПО

Замечания и
предложения

3

Инженер и программист
отрабатывают ПО на стенде и объекте

Объектный
код

- Длинный цикл разработки алгоритмического/программного обеспечения
- Глубокое вовлечение программиста в специфику работы конкретной системы

* специалист по конкретному объекту управления, контроля и т.п.



Разработка алгоритмического обеспечения (2)

Цикл разработки на базе комплекса РСДА

Оперирует понятиями:
сигнал, циклограмма,...

1

Инженер
разрабатывает алгоритм
работы системы
с помощью САПР*
(редактора алгоритмов)



2

Инженер
отрабатывает ПО на стенде
программного комплексирования

**Программист не участвует в
разработке, реализации и отработке
алгоритмического обеспечения**

3

Инженер
отрабатывают ПО на стенде** и объекте



* Система Автоматизированного Проектирования

** Стенд программно-аппаратного комплексирования и полунатурного моделирования



Разработка алгоритмического обеспечения (3)

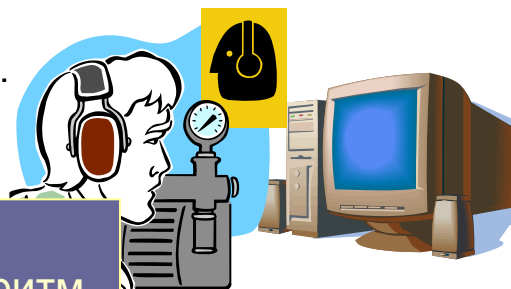
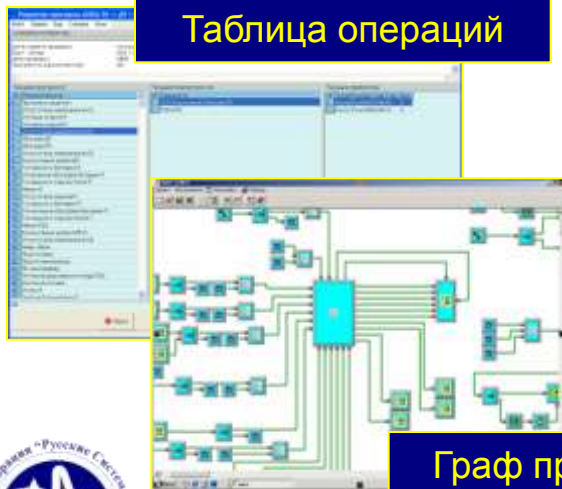
Разработка алгоритмов с помощью САПР

Оперирует понятиями:
сигнал, циклограмма,...

1

Инженер
разрабатывает алгоритм
работы системы
с помощью САПР*
(редактора алгоритмов)

Таблица операций



Эффективная разработка
алгоритма возможна при работе
инженера в понятных ему
терминах и понятиях

Структурированный инженерный язык

```
File Edit Format View Help
Модуль 500.112 "Запись N2np начала прикрытия сопла"
Действия по входу
Цикл контроля
    Если ((Aес_МГ_П - A_ГС) > 10) //Сопло прикрылось на 10 от МГ
        Команда Присвоить N2_ЗАКР_РС_П = N2npв8
        Команда Присвоить Д_ЗамеренЗАКР_РС = Д_ЗамеренЗАКР_РС + 1
        Команда Вызвать модуль контроля 500.113
        Команда Запустить модуль контроля 500.114
        Команда Завершить цикл контроля
    Если (Д_Режим = 0)
        Команда Завершить цикл контроля
Действия по выходу
Модуль 500.113 "Контроль N2np начала прикрытия сопла"
Действия по входу
    Команда Присвоить N2_ЗАКР_РС_ЗД_П = 0
Цикл контроля
    Если (abs (N2_ЗАКР_РС_П - N2_ЗАКР_РС_Э) > 2)
        Команда Присвоить N2_ЗАКР_РС_ЗД_П = 1
    Если 1
        Команда Завершить цикл контроля
Действия по выходу
```

Граф процесса



Разработка алгоритмического обеспечения (4)

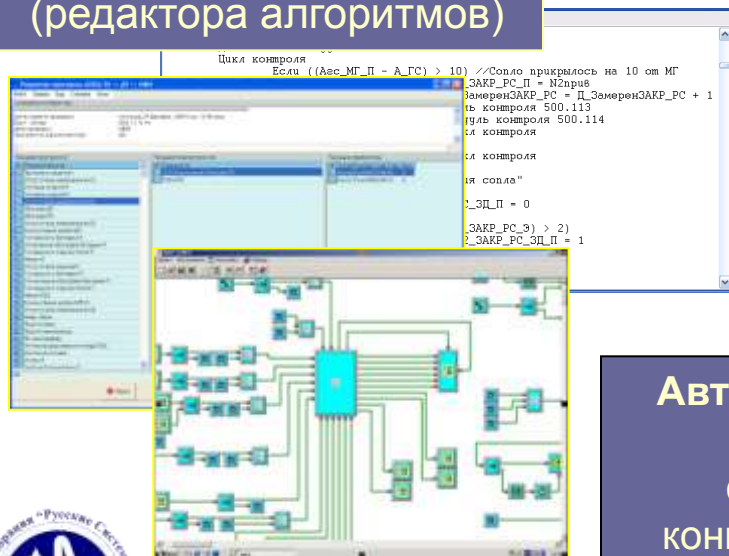
Разработка алгоритмов с помощью САПР

Опиерирует понятиями:
сигнал, циклограмма,...

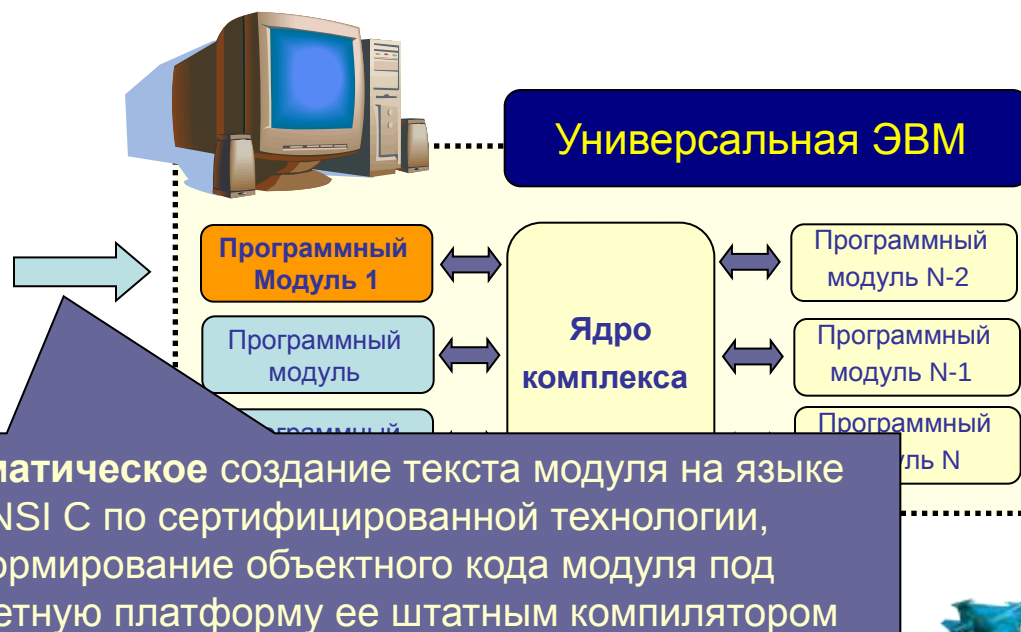


Инженер

разрабатывает алгоритм
работы системы
с помощью САПР
(редактора алгоритмов)

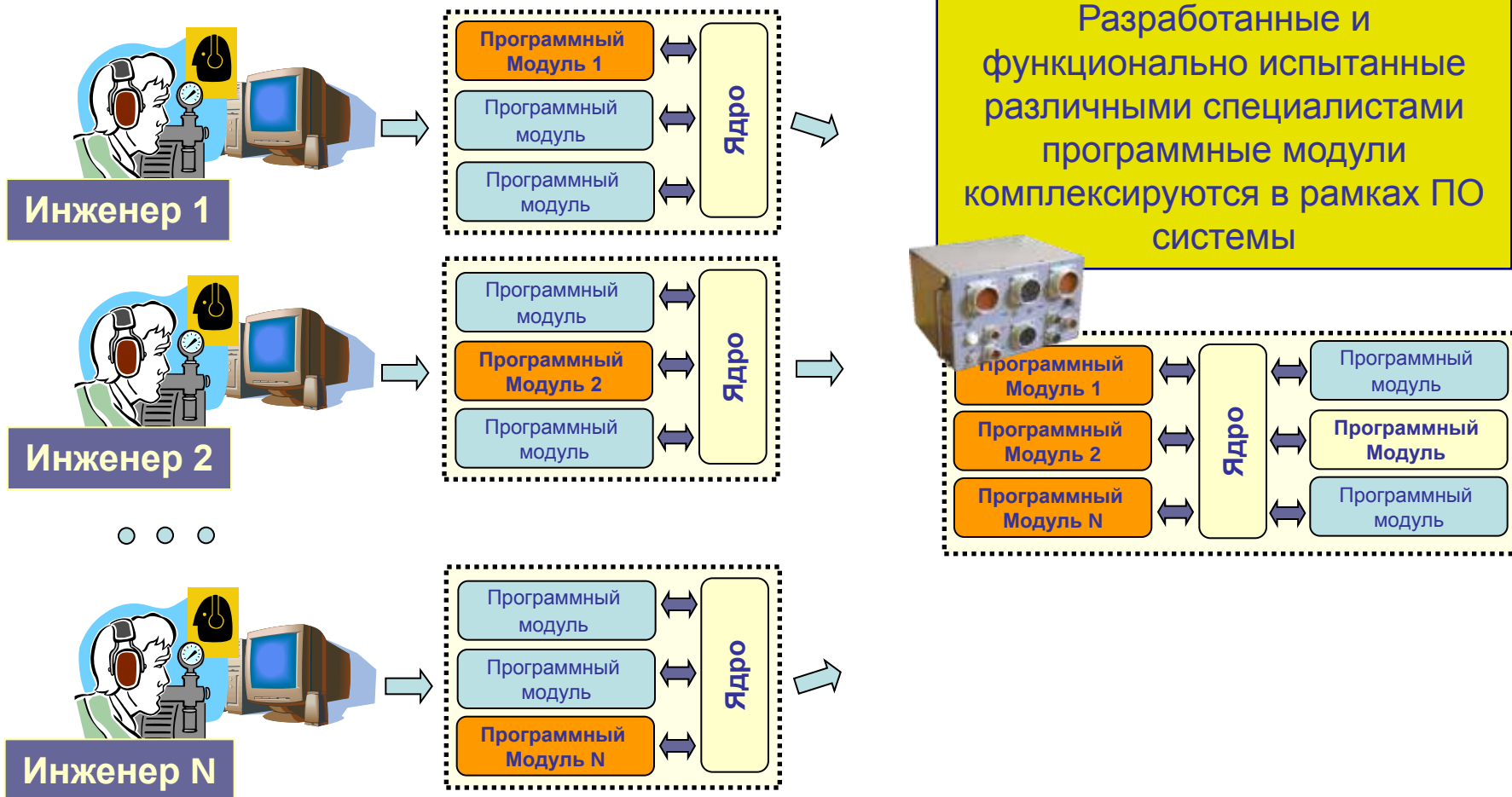


Разработанный алгоритм
автоматически преобразуется в
программный модуль комплекса и
отрабатывается на стенде
программного комплексирования



Разработка алгоритмического обеспечения (5)

Коллективная разработка комплекса



Разработка элементов отображения информации

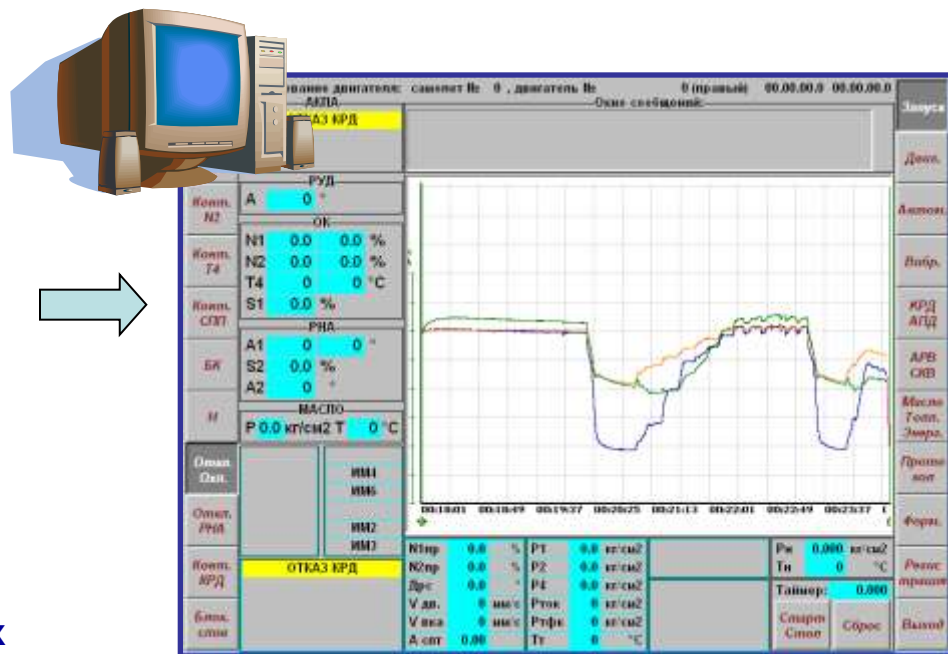
Оперирует понятиями:
график, таблица,...



Инженер имеет возможность оперативно
изменить объем и форму отображения
информации на дисплее

```
Файл Правка Формат Вид Справка
#ИМЯ "Выбор борта для работы"
#ЭКРАН W 1024 H 768
#ФОН С 8
#ШРИФТ 5 "Arial_RS1" 12 ["fsBold"] ["fsItalic"]
#ВКЛЮЧИТЬ ФАЙЛ "format.001.txt"
#БАЗА X 85 Y 126
#ТЕКСТ X 1 Y 19 W 200 H 18 S 0 "Бортовой номер"
#СП_ВЫБОРА X 1 Y 38 W 443 H 596 Cb 8 Cf 0 Ch 14
#КНОПКА X 0 Y 0 W 76 H 76 D 100000 РП_БПР
#КНОПКА X 0 Y 76 W 76 H 76 D 100000 РП_БПР
```

**Настройка элементов отображения
осуществляется в терминах, удобных
инженеру, в настроечном файле модуля
отображения информации**



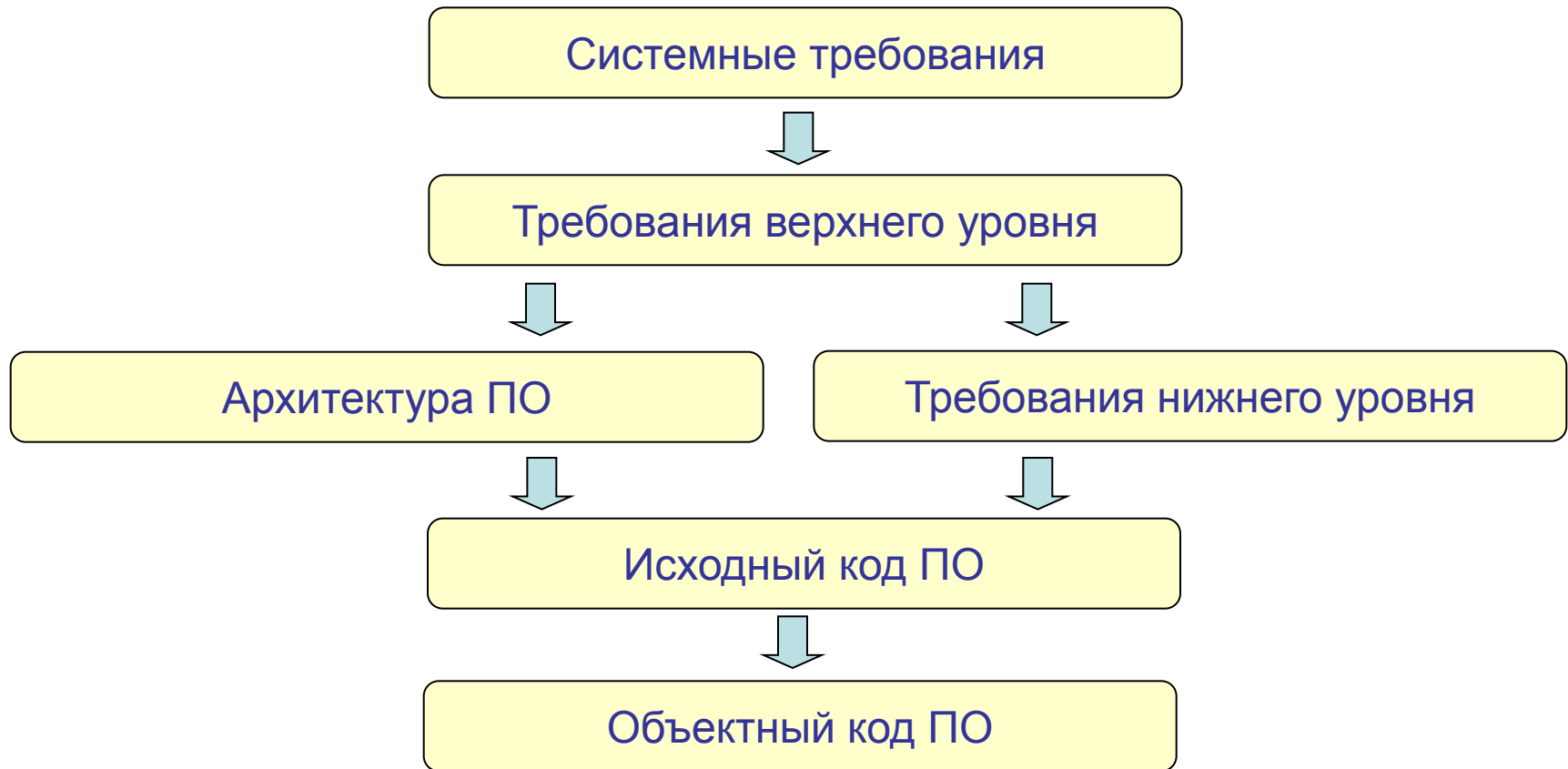


Инструментальные средства для анализа и поддержки требований и верификации ПО на основе формализованных требований

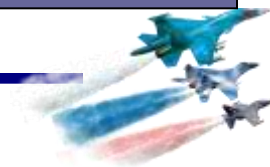
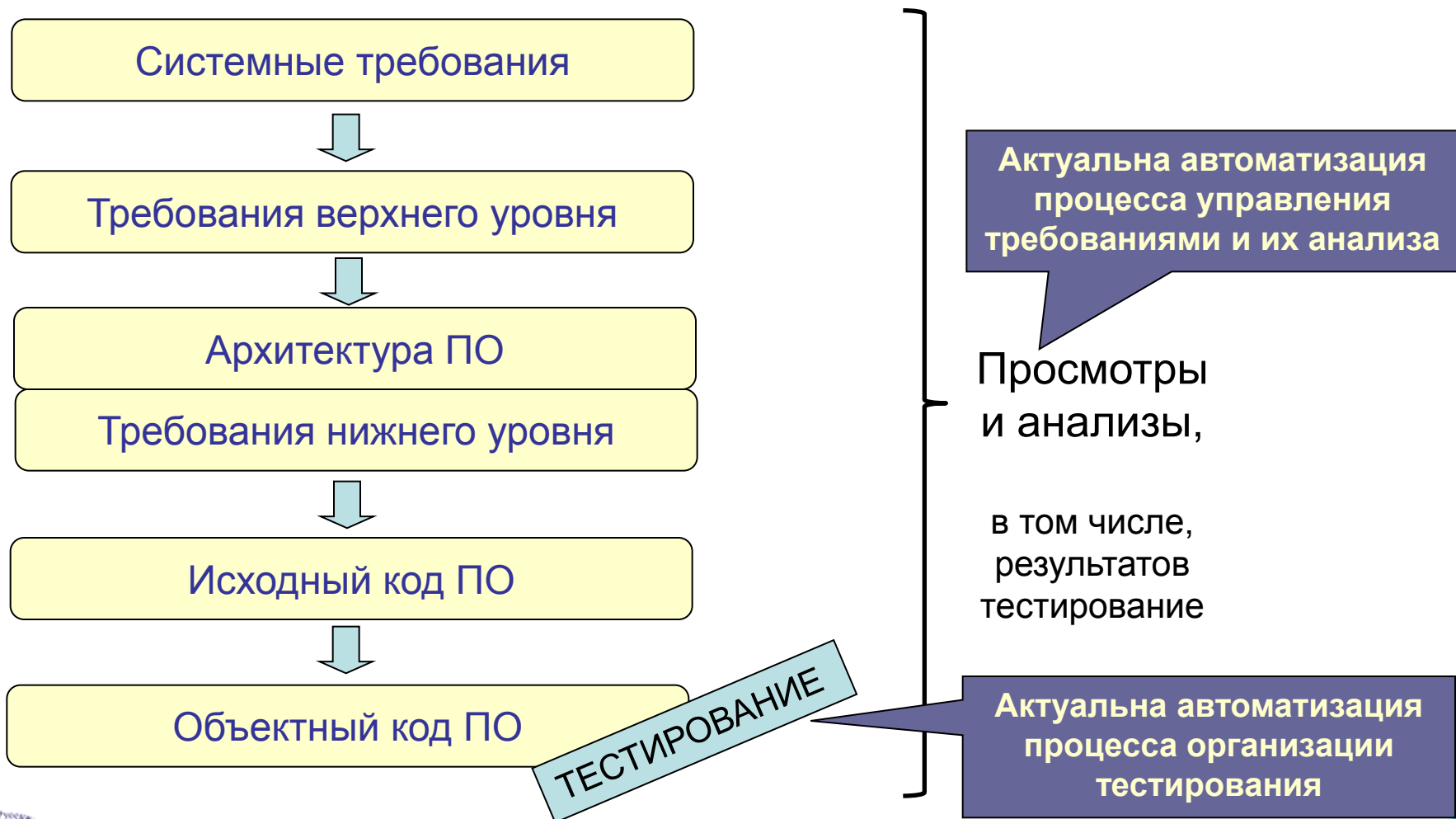


Цели верификации программного обеспечения

Цель верификации ПО – обнаружить и зарегистрировать ошибки, внесенные в ПО на всех этапах его разработки



Методы верификации программного обеспечения





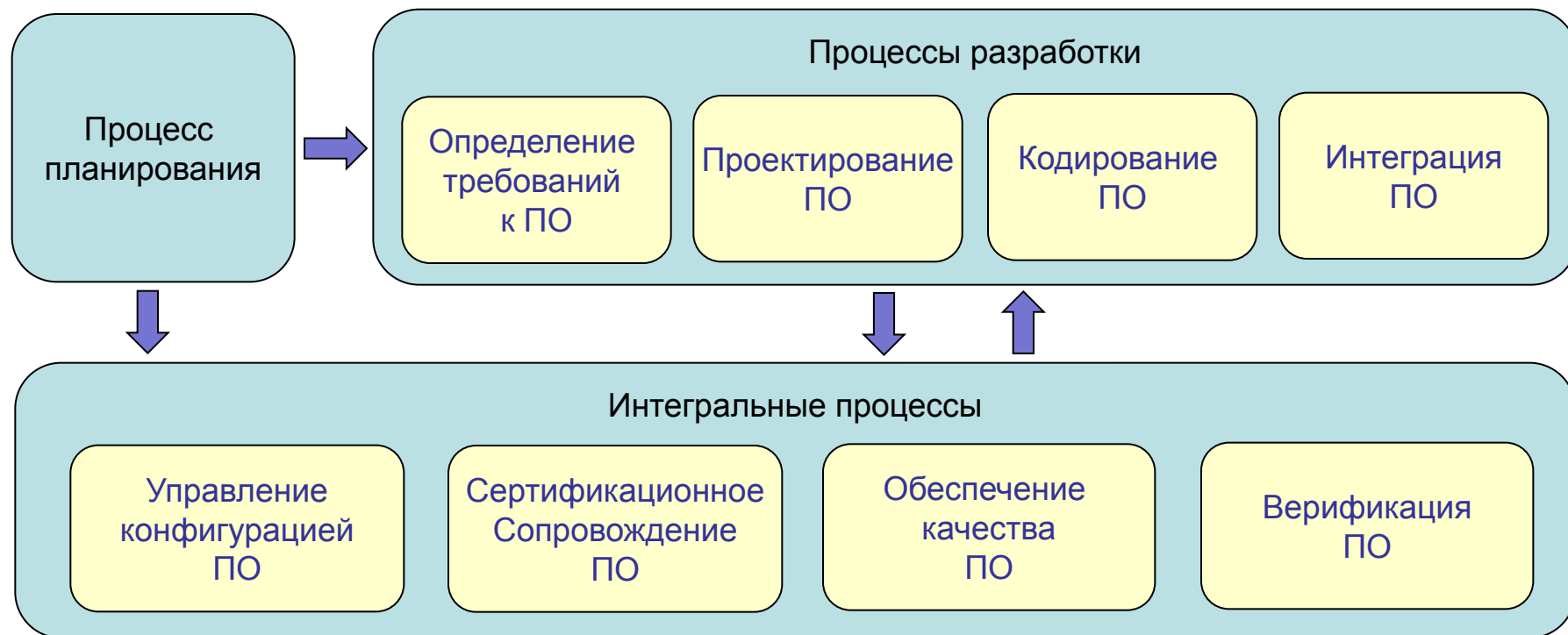
Автоматизированная система управления процессами (АСУП) жизненного цикла программного обеспечения



Типовые процессы жизненного цикла ПО

ГОСТ Р 51904

Необходимо эффективно осуществлять комплексное планирование и контроль работ в рамках жизненного цикла ПО и соответствующих процессов



Документальное сопровождение

Применение технологии разработки ПО на базе открытого кросс-платформенного программного комплекса РСДА позволяет значительно **снизить затраты на документирование** процессов разработки в соответствии с действующими стандартами:

- ЕСПД
- ГОСТ Р 51904-2002
- ГОСТ РВ 0019-001-2006
- и др.



за счет использования технологий поддержки жизненного цикла ПО, типовых форм документов, а также разработанных документов повторно используемых программных модулей



Результаты использования

Опыт практического применения представленных
методик, базового и инструментального (технологического) ПО
в ОАО «Корпорация «Русские системы» обеспечило



Многokратное снижение затрат
(стоимости и сроков) на разработку
программного обеспечения



Повышение уровня безопасности
программного обеспечения (снижение
вероятности наличия ошибок)





ОАО «Корпорация «Русские системы» заинтересовано в
распространении и апробации представленных методов и средств
в широком спектре систем специального назначения

www.rusys.ru