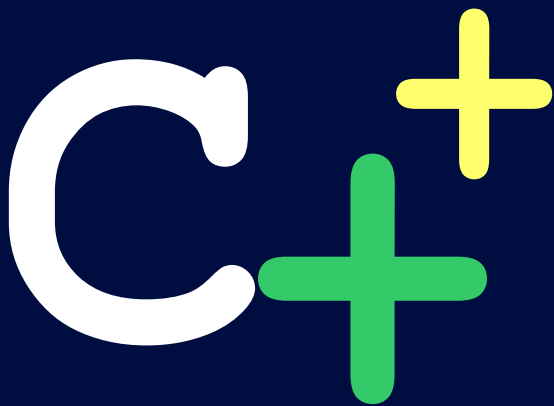




Создание среды разработки для C++

Профессиональная
конференция
для embedded-
разработчиков C++

Взгляд изнутри



CM-LYNX. Создание отечественной среды разработки для DSP процессоров

Аналитик отдела технологий разработки
интеллектуальных систем
Центра проектирования машинно-обучаемых
доверенных интеллектуальных систем

Юлия Смирнова

Великий Новгород

01.06.2019

Оглавление

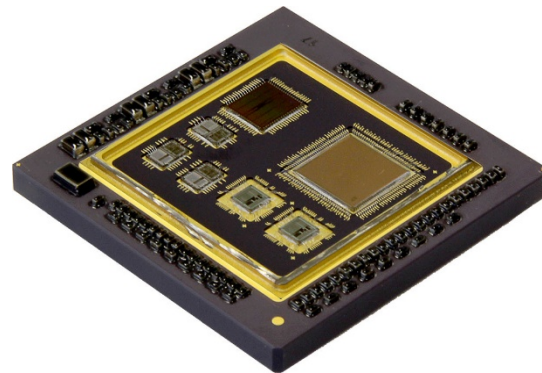
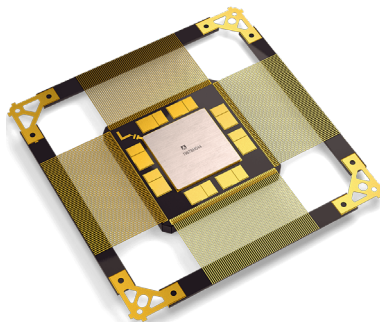
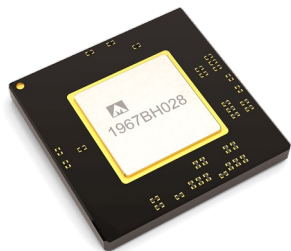
- 01. CM-LYNX - отечественная среда разработки для DSP процессоров
- 02. Проблемы и решения
- 03. Подробнее о процессах и команде
- 04. Достижения, планы развития

01.

СМ-LYNX -отечественная среда разработки для DSP процессоров

ВПЦОС 1967ВН028 и 1967ВН044

- 32-разрядные высокопроизводительные процессоры цифровой обработки сигналов
- Ближайший аналог ADSP-TS201



Собственные средства разработки

- Оригинальные архитектура ядра и система команд процессора
- Кроссплатформенность
- Возможность сертификации ФСТЭК



Состав интегрированной среды обработки CM-LYNX

- Графическая оболочка
- Компиляторы C/C++
- Компилятор ассемблера
- компоновщик
- Отладчик
- Программный симулятор
- Средства постобработки бинарных образов
- Программные библиотеки:
 - Библиотека цифровой обработки
 - HAL
 - Операционная система реального времени МАКС*

* Разработка МИР



Ранние версии. Пробы и ошибки

- Отсутствие опыта разработки масштабных программных продуктов
- Обширная заявленная функциональность:
 - Замена VDSP с сохранением привычного программного интерфейса
 - Программный симулятор
 - Версии для Windows и Linux
 - Операционная система реального времени
 - Графические средства RAD
- Разработка продукта силами в сложной кооперации
- Отсутствующая функциональность
- Низкое качество генерируемого кода
- Отсутствие единого продукта и процесса его производства
- Редкие обновления
- Ошибки, ошибки, ошибки...

FAIL

Работа над ошибками

С апреля 2018 г.



Рабочая группа



Процесс выпуска версий

- Управление требованиями
- Стандарты качества
- Интеграционное и системное тестирование
- Вторая линия технической поддержки
- Управление распределенными командами



Дорожная карта
продукта



Команда
разработки

02.

Проблемы и решения

Как все начиналось



- Новая команда
- Отсутствие опыта
- Отсутствие регламентов
- Отсутствие общей инфраструктуры
- Багаж «багов»
- Отсутствие критически важного функционала
- Неактуальная и неполная документация

Исправление дефектов: начальные условия

На вход:



- Список «багов» в виде Excel неизвестной степени актуальности (порядка 70шт.)
- Сообщения от службы техподдержки, иногда дублирующие «баги» из Excel
- Сообщения в духе «все сломалось»

Проблемы:

- Не понятна суть «бага»
- Нет связи с пользователем
- Не известна актуальность «багов» - исправлены или нет
- Не понятна критичность для пользователя

Исправление дефектов: что было сделано

1. Настроена инфраструктура

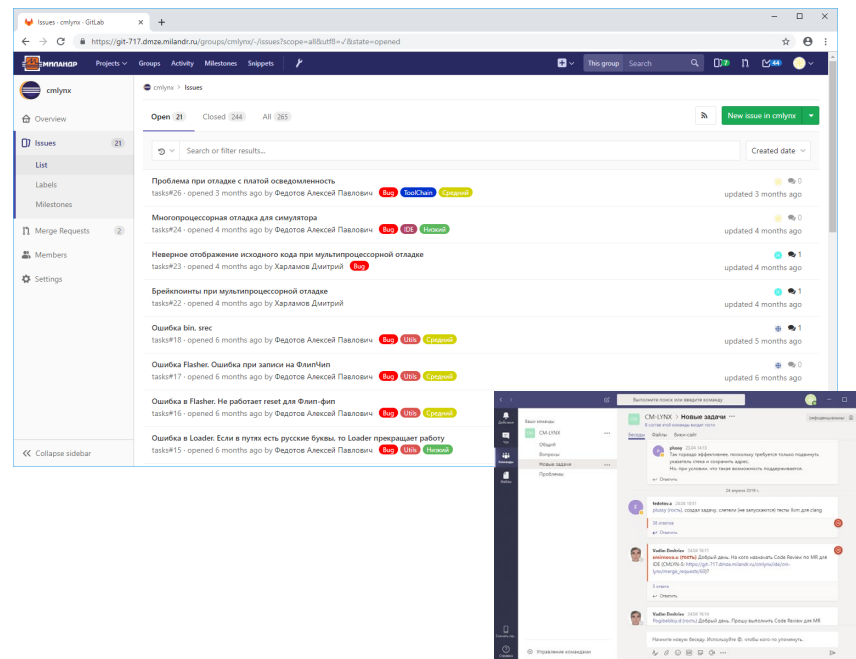
- Gitlab для хранения и совместной работы с кодом
- Issue Manager Gitlab как система управления задачами (сейчас JIRA&Confluence)

2. Регламентированы процессы

- Регламенты
- Вторая линия техподдержки
- Управление распределенной командой

3. Систематизированы дефекты

- Классификация, приоритезация дефектов
- Задачи в Issue Manager Gitlab



Отсутствие критически важного функционала: проблемы и решения

Проблемы:

- ✓ Отсутствие функционала, который есть в VDSP
- ✓ Отсутствие многопроцессорной асимметричной отладки
- ✓ Часть якобы реализованного функционала работает некорректно или вообще отсутствует

Решения:



Договоры на разработку

Реализация нового функционала командой сопровождения

Разработка нового функционала по договору



- + Фиксированная стоимость работ
- + Прописанные обязательства
- + Независимая работа команды
- Бюрократия
- Разногласия вокруг ТЗ
- Сложность внесения изменений в требования
- О проблемах узнаем в конце проекта
- Сложность интеграции результатов

Разработка нового функционала командой



- + Работа без ТЗ, простота внесения изменений в требования (Agile)
- + Есть возможность декомпонировать задачу и раньше поставить результат пользователю
- + О проблемах узнаем своевременно
- + Проще интеграция результатов
- Стоимость работ может быть выше
- Не зафиксированы обязательства

Разработка нового функционала



Новый функционал - независимый проект
или
совместная работа «распределенной» команды
над развитием продукта?

**Выбор - совместная работа
«распределенной» команды над развитием
продукта!**

Документация: проблемы и решения

Проблемы:

- Неполная и неактуальная пользовательская документация
- Отсутствие единого стиля оформления
- Практически полное отсутствие проектной документации



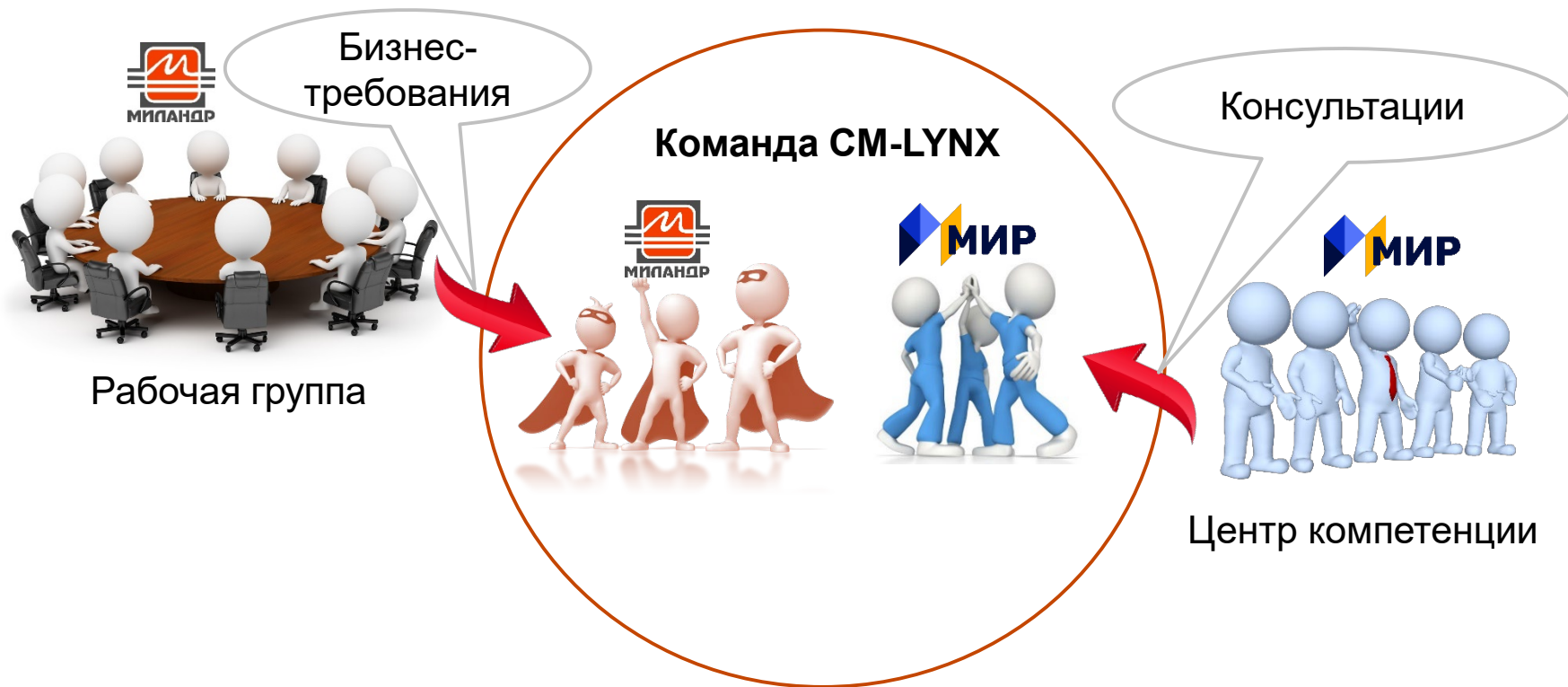
Что было сделано:

- Комплект пользовательской документации в едином стиле
- Процесс обновления документации перед каждым релизом
- Работа по созданию проектной документации
- Единая база знаний по продукту в Confluence

03.

Подробнее о процессах и команде

Наша команда



Процесс исправления дефекта



Процесс разработки нового функционала

1.



Проектирование, оценка

3.



Разработка, тестирование

2.



Презентация решения и выбор варианта

4.



Код-ревью, тестирование

Процесс выпуска новой версии



Сборка компонент

2.



Тестирование



Документирование

4.



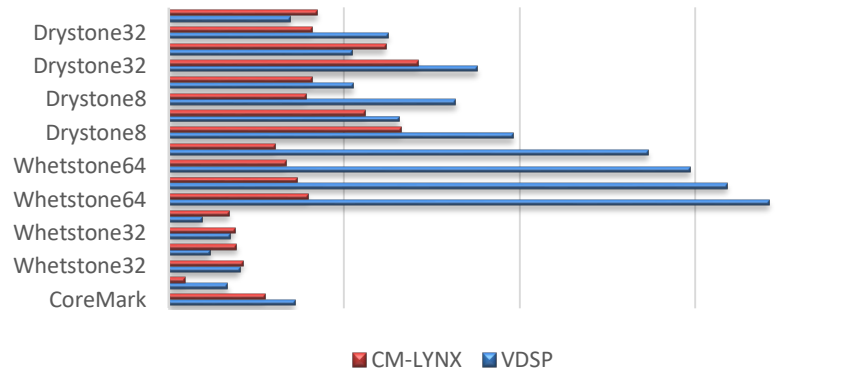
Распространение

04. Достижения, планы развития

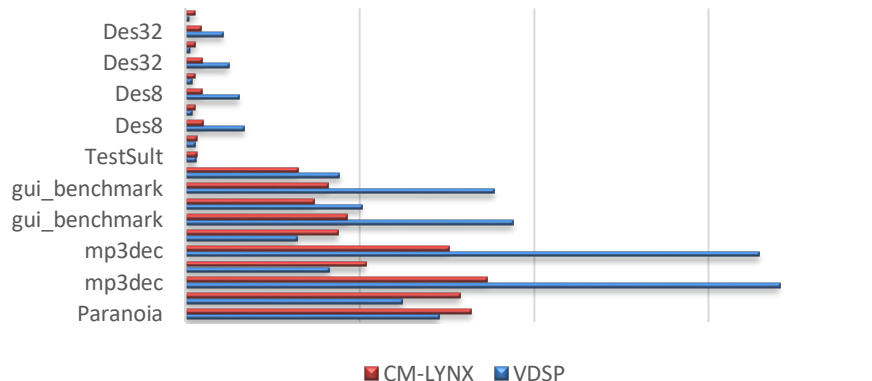
Результаты

- Выпущено 7 версий (с апреля 2018г.)
- Актуализирована документация
- Повышена стабильность:
 - Исправлено 330 дефектов
- Реализован новый функционал:
 - Библиотека DSP
 - Builtin-функции
 - Переработан режим CHAR8
 - Программный симулятор
- Производительность кода сравнима с VDSP, иногда лучше

Время выполнения



Время выполнения



Планы развития CM-LYNX



- Новый отладчик
 - Многопроцессорная симметричная и ассиметричная отладка
 - Удаленная отладка
- Оптимизация
 - Полноценное использование набора команд
 - Оптимизация во время компоновки
- Поддержка новых типов ВПЦОС Миландр
 - Поддержка системы команд и возможностей
- Совместимость с VDSP
 - Более полная совместимость с кодом из VDSP
 - Утилиты миграции
- Версия для ОС Linux
- Обучающий курс для пользователей

Спасибо!

