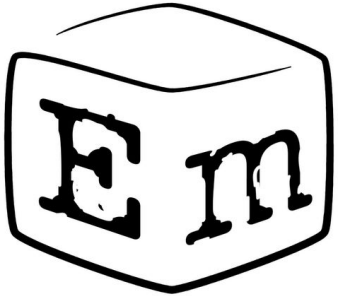


СПО в процессах безопасной разработки на примере ОСРВ Embox

Антон Бондарев

OSSDEVCONF, 2023

Embox

Embox — свободная операционная система реального времени (RTOS), разрабатываемая для встроенных систем

Основная идея Embox использование ПО Линукс в более безопасном и детерминированном, менее ресурсоемком и энергопотребляющем окружении.

Проблема OpenSource

- Более 95% ИТ-компаний в мире сейчас используют opensource-решения.
- Но считается что СПО менее надежно и безопасно, если речь идет об промышленном ПО, где требуется предсказуемость поведения.

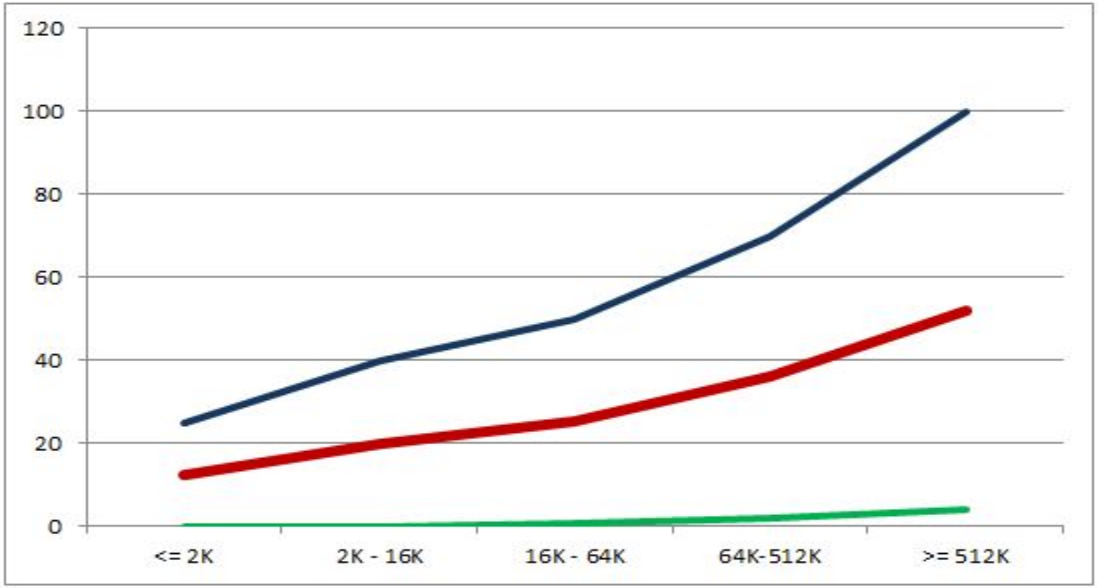
Методы повышения надежности

- Статические анализаторы
- Использование подходящих архитектурных решений
- Средства разработки
- Использование тестирования
- Языки высокого уровня
- Подходящие процессы разработки

Статические анализаторы

- Количество ошибок на 1000 сток кода — показатель надежности программного обеспечения. Стандартом для промышленного ПО определен, не больше 1. Разный для разных проектов, никогда не равен 0.

Количество ошибок



Opensource vs. Proprietary

TABLE 7: 2013 DEFECT DENSITY BY PROJECT SIZE OPEN SOURCE VS. PROPRIETARY C/C++ CODE		
Size of Codebase (Lines of Code)	**Open Source**	**Proprietary Code**
Less than 100,000	.35	.38
100,000-499,999	.50	.81
500,000-1 million	.70	.84
More than 1 million	.65	.71
Average across projects	.59	.72

Opensource

Table B: Key Findings for Linux 2.6, PHP 5.3, and PostgreSQL 9.1			
	Linux 2.6	**PHP 5.3**	**PostgreSQL 9.1**
Lines of code scanned	6,849,378	537,871	1,105,634
Defect Density (as of 12/31/11)	0.62	0.20	0.21
Number of outstanding defects (as of 12/31/11)	4,261	97	233
Number of defects fixed in 2011	1,283	210	78
Number of outstanding defects (as of 1/1/11)	3,457	14	247

Архитектуры ОС

- Микроядерная
- MILS
- Монолитная
- …

Архитектуры не обеспечивают характеристики в общем случае. Различные подходы эффективны для решения различных задач. Идеально, если есть возможность задавать комбинацию свойств в зависимости от задачи.

Средства разработки

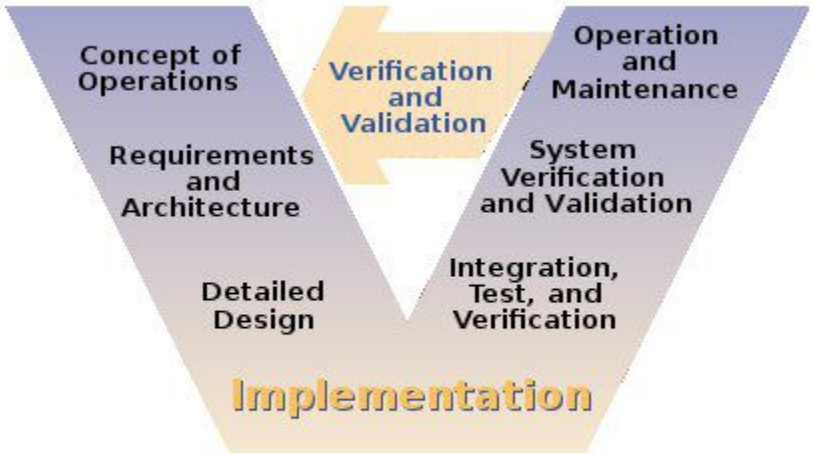
- Санитайзеры
- Флаги компиляции
- Флаги оптимизации

Тестирование

- Unit тестирование
- фазинг
- функциональное тестирование
- нагрузочное тестирование
- Регрессионное тестирование
- Интеграционное тестирование

Процессы разработки

DO-178 (КТ-178) стандарт на процесс разработки



Evaluation Assurance Level (EAL)

- EAL1: Functionally Tested
- EAL2: Structurally Tested
- EAL3: Methodically Tested and Checked
- EAL4: Methodically Designed, Tested, and Reviewed
- EAL5: Semiformally Designed and Tested
- EAL6: Semiformally Verified Design and Tested
- EAL7: Formally Verified Design and Tested

Языки высокого уровня

- Языки с управляемой памятью
- Языки с высокой типизацией и статической проверками
- Специализированные языки (Erlang)
- Функциональные языки
- DSL языки (для дополнительной информации)

СПО

- Высокая переиспользуемость кода
- Большое количество разнообразных ситуаций
- Распределенные (поддерживаемые) решения
- Возможность применять анализ кода и тестирование на ранних стадиях проекта

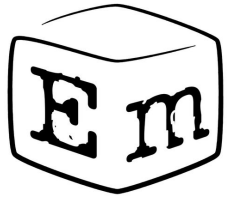
Embox

- Continius Integration
  - Unit тестирование
  - Интеграционное тестирование
- Профилировщики
- Санитайзеры
- Статические анализаторы
  - CoverityScan
- DSL язык (Mybuild) для описания модулей и требований к системе
- Возможность использовать СПО

Итоги

- **СПО** увеличивает надежность и безопасность конечных систем за счет широкого использования в различных ситуациях
- Для получения эффекта от СПО необходимо выстраивать **процессы разработки**
- Средства заложенные в **Embox** позволяют выстраивать качественные процессы разработки для построения надежных и безопасных систем

Контакты



Страница проекта

http://embox.github.io/

Репозиторий проекта

https://github.com/embox/

Антон Бондарев

anton.bondarev2310@gmail.com