Аппаратное обеспечение IoT/CPS
Лекция 10

А. А. Подшивалов
apodshivalov@miem.hse.ru

Функциональная безопасность встраиваемых систем

Найдите лишнее

Критическая информационная инфраструктура

Ответственность

I. Уголовный кодекс Российской Федерации:

Статья 217.1. Нарушение требований обеспечения безопасности и антитеррористической защищенности объектов топливно-энергетического комплекса

Статья 272. Неправомерный доступ к компьютерной информации

Статья 273. Создание, использование и распространение вредоносных компьютерных программ

Статья 274. Нарушение правил эксплуатации средств хранения, обработки или передачи компьютерной информации и информационно-телекоммуникационных сетей

Статья 274.1. Неправомерное воздействие на критическую информационную инфраструктуру Российской Федерации

Статья 283. Разглашение государственной тайны

Статья 284. Утрата документов, содержащих государственную тайну

Статья 293. Халатность

Немного определений (по ГОСТ Р 56205–2014, он же IEC/TS 62443-1-1:2009)

- Безопасность (safety) — отсутствие *недопустимого риска*

Немного определений (по ГОСТ Р 56205–2014, он же IEC/TS 62443-1-1:2009)

- Безопасность (safety) — отсутствие *недопустимого риска*
- Защита (security) — предотвращение несанкционированного или нежелательного проникновения, а также вмешательства в исправную и запланированную работу системы промышленной автоматики и контроля (одно из определений)

Немного определений (по ГОСТ Р 56205–2014, он же IEC/TS 62443-1-1:2009)

- Безопасность (safety) — отсутствие *недопустимого риска*
- Защита (security) — предотвращение несанкционированного или нежелательного проникновения, а также вмешательства в исправную и запланированную работу системы промышленной автоматики и контроля (одно из определений)
- Кибербезопасность (киберзащита) (cybersecurity) — действия, необходимые для предотвращения неавторизованного использования, отказа в обслуживании, преобразования, рассекречивания, потери прибыли, или повреждения критических систем или информационных объектов

Допустимые и недопустимые риски

- Опасность (hazard) — потенциальный источник причинения вреда
- Вред (harm) — физическое повреждение или ущерб, причиняемый здоровью людей, имуществу или окружающей среде
- Риск (risk) — сочетание вероятности события причинения вреда и тяжести этого вреда
- Допустимый риск (tolerable risk) — риск, который приемлем при данных обстоятельствах на основании существующих в обществе ценностей

Немного философии, или об оценке рисков

- ALARP (as low as reasonably practicable)
- GAMAB (globalement au moins aussi bon)
- MEM (minimum endogenous mortality)

И еще немного определений

- Ошибка (error) — применительно к ПО — ошибка в требованиях, проекте или коде
- Неисправность (fault) — состояние системы, когда она не соответствует требованиям нормативной или конструкторской документации; применительно к ПО — проявление ошибки в программном обеспечении
- Отказ (failure) — неспособность системы выполнять требуемую функцию (*возможно*, в результате проявления неисправности)

Основные стандарты

- IEC 61508 — «Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью»
  - ISO 26262 — Автомобильный транспорт
  - EN 50126, EN 50128, EN 50129 — Железнодорожный транспорт
  - IEC 62061 — Системы управления
  - IEC 62304 — Медицинские приборы
  - IEC 62443 — Промышленная автоматика
  - DO-178C — Авиационная техника

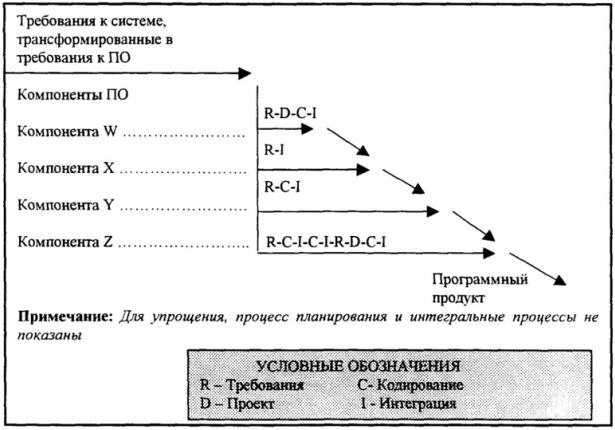
Разработка безопасных систем

Основные «документы»

- Анализ опасностей и рисков (hazard and risk analysis)
- Анализ отказов (failure analysis)
  - FMEA — анализ видов и последствий отказов
  - FTA — анализ дерева отказов
- Обоснование безопасности (safety case)
- План обеспечения безопасности (safety plan)
- Руководство по безопасности (safety manual)

Не обязательно «водопад»

Рис. 3-1 иллюстрирует последовательность процессов при разработке нескольких компонент одного программного продукта, имеющих различные жизненные циклы.



Некоторые противоречия

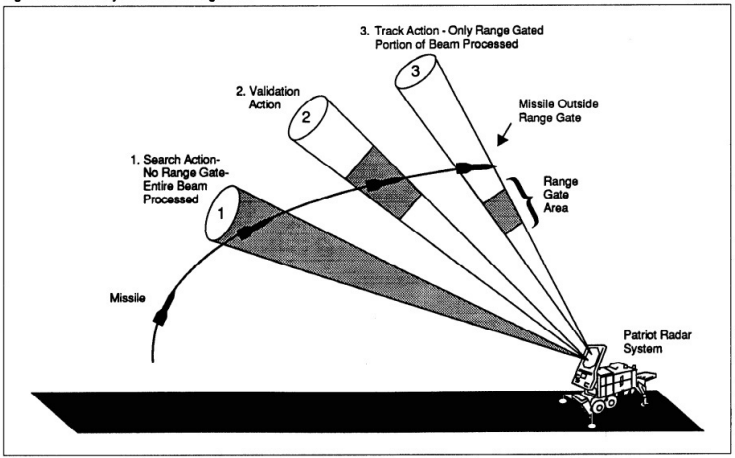
- Доступность/надежность
  - Может ли система выдать неправильный отклик, но вовремя?
- Функциональность/безопасность
  - А если ничего не делать...
- Защищенность/производительность/безопасность
- Уровень полноты безопасности (SIL, safety integrity level)

Обнаружение ошибок

- Можно ли доверять внешним данным?
  - Что делать, если данные «неправдоподобны»?
- Аномалии во внутренних метриках системы
- Rejuvenation
  - Накопление ошибок
  - Patriot 25.02.1991

Patriot 25.02.1991

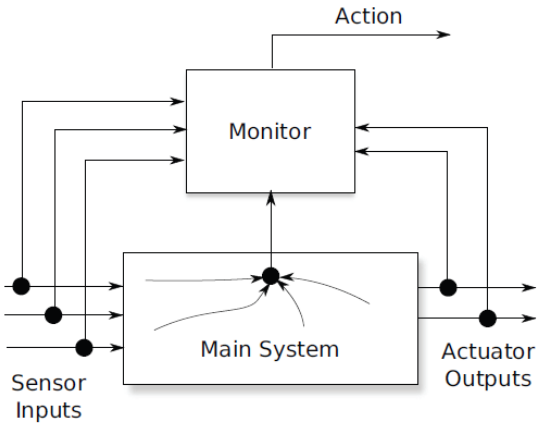
Figure 5: Incorrectly Calculated Range Gate



Обнаружение ошибок — что делать?

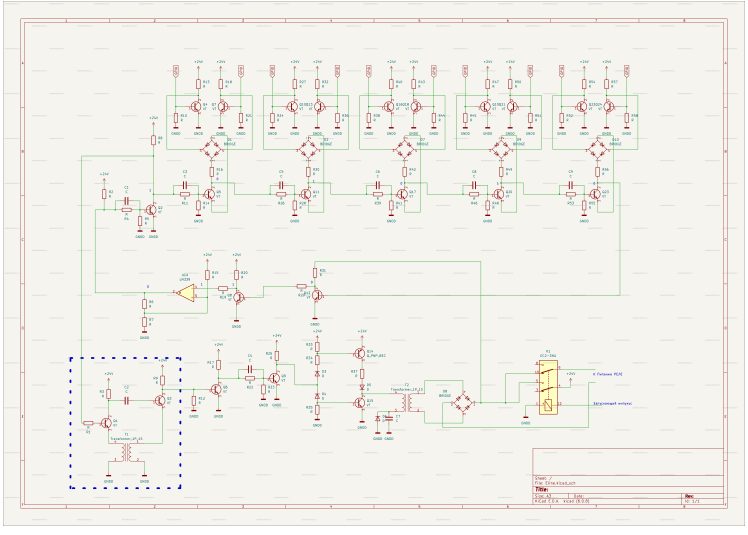
- Заведомо безопасное состояние
- Восстановление системы
- Fail-fast, crash only и тому подобные подходы — максимизация ошибки

Дублирование и мажорирование



- Имеет смысл, если надежность ПО намного превышает надежность аппаратной части
- Вырожденный случай 1 — многоверсионное разнородное ПО: два процессора разной архитектуры, две команды программистов, ...
- Вырожденный случай 2 — watchdog

Давайте включим реле...



Программное обеспечение

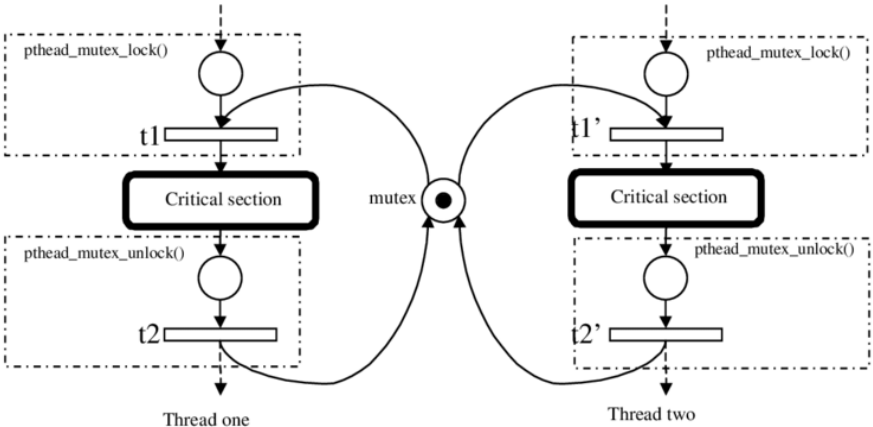
Процессы жизненного цикла ПО

- Планирование
  - Средства разработки
    - MISRA C — подмножество языка C
    - Ada — язык с поддержкой некоторых формальных методов верификации
  - Среда испытаний
- Проектирование
  - Прослеживаемость требований
- Кодирование
- Интеграция
- Верификация
  - Тестирование, анализ покрытия тестами
  - Статический анализ

Некоторые вопросы верификации

- Отказы ПО — детерминированные или случайные?
- Надежность ПО
- Верификация относится не только к исходному или объектному коду, но и к требованиям и проекту
  - Полуформальные методы — конечные автоматы, сети Петри, ...
  - Формальные методы — pre-conditions, post-conditions, инварианты
  - Инструментальные средства верификации — от симуляторов до систем доказательства теорем

Mutex в нотации сети Петри



Требования к средствам разработки

- Обычно средства разработки должны быть *квалифицированы* в соответствии с используемым стандартом
- То же самое относится и к используемым сторонним компонентам
- SOUP — Software of Unknown Provenance
  - Разработка в соответствии с IEC 61508
  - PIU — proven-in-use
  - Оценка не соответствующего стандарту ПО