Embedded systems engineering
Distributed real-time systems

David Kendall

Introduction

- Version Control Systems
  - Mercurial
- C Programming Standards
  - MISRA
- Development Methods
  - Agile

Version Control Systems

- Version Control Systems are used to keep a record of the history of the development of a project (source code, assignment, dissertation, . . .)
- Project history is stored in a repository
- You can revert to any previous project state, share changes with others, create a new branch and merge changes when required
- VCS may be centralised
  - Subversion
- or distributed
  - Mercurial
  - Git
- Focus here is on Mercurial

Mercurial

from http://mercurial.selenic.com

“Mercurial is a free, distributed source control management tool. It efficiently handles projects of any size and offers an easy and intuitive interface.”

- Open source, written in Python, available for Unix, Windows, Mac, . . .
- Widely used, e.g. Mozilla, OpenOffice, Python, W3C (more)
- Get started now
- Understanding Mercurial gives a more detailed view of working with Mercurial repositories
- Joel Spolsky has a nice tutorial at http://hginit.com/
- Most Windows users work with TortoiseHg

MISRA C

- The official MISRA C website
- Guidelines for the use of the C programming language in safety critical systems produced by the Motor Industries Software Reliability Association
- Widely accepted model of best practice in many sectors: aerospace, telecom, medical devices, defense, railway, ...
- First edition in 1998, second edition in 2004, third edition in 2012
- 2012 edition has 143 rules and 16 directives, classified as “mandatory”, “required” or “advisory”
- Unfortunately not free to users or implementors
- See this Wikipedia entry for a more detailed introduction

Agile Development Methods

- A (better) alternative to a waterfall development method
- Propounded by Kent Beck and others in the Agile Manifesto
- Incremental approach to development
  - Develop software early
  - Build incrementally
  - Have a test plan from the start; perform regression tests
  - Refactor – constantly assess the design, be prepared to change in order to enhance essential qualities: low coupling, high cohesion
- Fits well with agile modelling
  - Develop models early
  - Model incrementally
  - Have a specification from the start; perform verification regressively
  - Refactor