FIRMWARE (AND HARDWARE) CONFIGURATION MANAGEMENT

Michael Melamed
Software/firmware Architect, HP Indigo Division
24-May-2011



 © Copyright 2011 Hewlett-Packard Development Company, L.P.

PRESENTATION HIGHLIGHTS

- Firmware (and hardware) – align on terminology
- The legitimation - why should we bother at all
- Content to manage – what is in the box
- Tools, means and methodologies – a very brief overview



 © Copyright 2011 Hewlett-Packard Development Company, L.P.

WHAT IS FIRMWARE (AND HARDWARE)

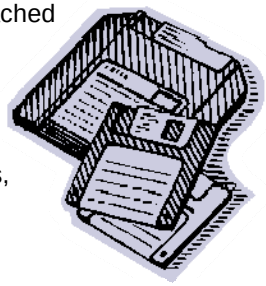
Wikipedia Definition (http://en.wikipedia.org/wiki/Firmware)

"*Today, in some circles, the word firmware has evolved to sometimes even denote application-level programs stored in NAND flash or NOR flash memory (such as in mobile phones and similar products). However, a more fundamental definition would still be fixed or semi-fixed data in a hardware device. This may include ROM and/or PLA structures for microcode and other data in a processor implementation as well as the low level machine code stored in ROM or flash memory running on that processor (i.e. many products use several levels of firmware). Microcode and other data in an application-specific integrated circuit (ASICs) would also fit that definition very well. The same could be said about programmable logic devices which may have configuration data stored either as internal fuses, in a ROM, or in a flash memory (used much the same way as a ROM or EPROM).*"

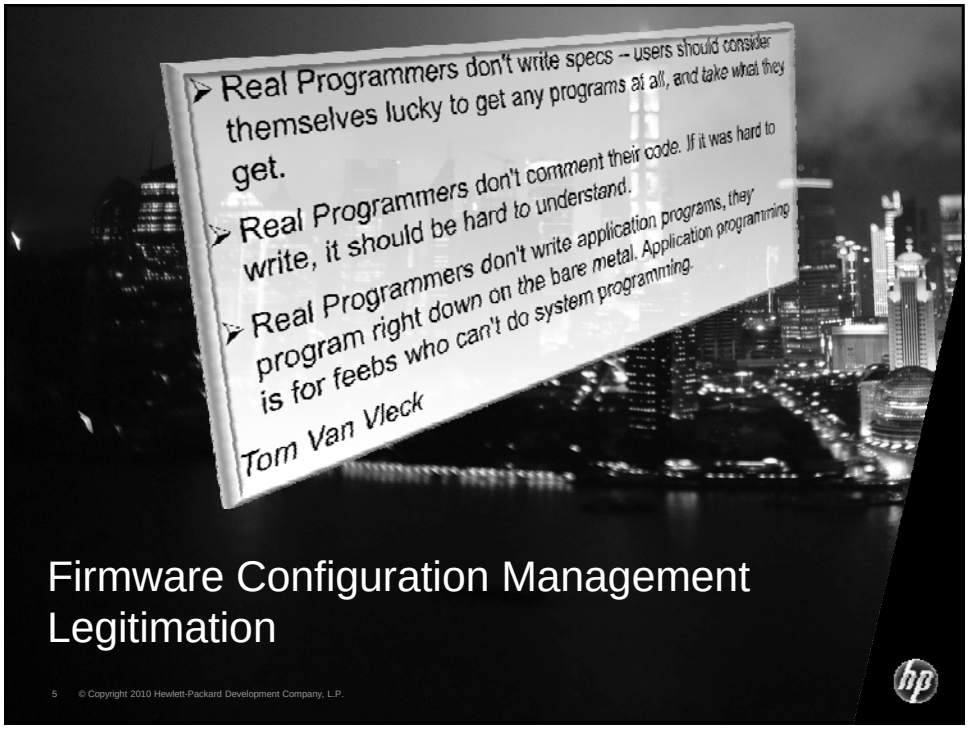
 © Copyright 2011 Hewlett-Packard Development Company, L.P.

FIRMWARE AREA OF RELEVANCE

- Personal computers and consumer peripherals (printers, scanners, disks, etc.)
- IT appliances (servers, routers, modems, network-attached storage, etc.)
- Telecom
- Control systems
- Consumer products (multimedia players, smartphones, PDAs, PNAs, digital photo cameras, etc.)
- Aircraft and airspace
- Automobiles
- Home appliances (dishwashers, washing machines, TV sets and boxes, smart home)
- Etcetera, etcetera



 © Copyright 2011 Hewlett-Packard Development Company, L.P.



Firmware Configuration Management Legitimation

 © Copyright 2010 Hewlett-Packard Development Company, L.P.

POPULAR ILLUSIONS AND EXCUSES



- We have no branching thus no configuration management is necessary
- We are really small team, so we can manage it by primitive means
- We do not have time/resources/budget for the configuration management

 © Copyright 2011 Hewlett-Packard Development Company, L.P.

THE REALITY

- Every product evolution sooner or later introduces the development stream branching:
  - Product lifecycle – code freezes, betas, formal versions, install base support, etc.
  - Hardware end-of-life
  - Product customizations
  - Cost reduction
  - Development tools and infrastructure upgrades
- New projects often inherit from the existing implementation
- Even in a very small team development is done in parallel.
- Improperly managed configuration results in significant time waste and actual expenses:
  - Customer support
  - Team multi-tasking
  - Research and concept development



 © Copyright 2011 Hewlett-Packard Development Company, L.P.

METHODOLOGICAL CONFIGURATION MANAGEMENT OBVIOUS ADVANTAGES

- Essential files are backed up
- Development process scalability
- Reversibility of the changes
- Simpler code re-use
- Proper interface declaration and handling
- Better control over the development cycle (changing tasks order)
- Appropriate customer support (reproduce-investigate-fix)



 © Copyright 2011 Hewlett-Packard Development Company, L.P.

WHAT CONTENT SHOULD BE MANAGED?

- Software source code
- Hardware logic design (VHDL)
- Hardware schemas (CAD projects and databases)
- Documentation
- Infrastructure and 3rd party components (RTOS, Open Source components, off-the-shelf libraries, etc.)
- Development environment and configurations (Eclipse, Wind River Tornado/Workbench, etc.)
- Debugging accessories operation (In-circuit emulators, logic analyzers, scopes, etc.)



 © Copyright 2011 Hewlett-Packard Development Company, L.P.

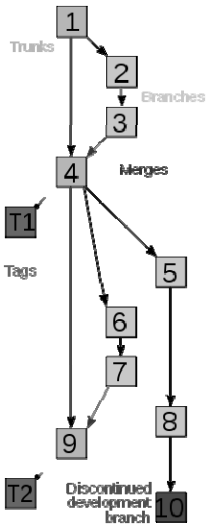
TOOLS, MEANS AND METHODOLOGIES

REVISION CONTROL SOFTWARE

The client-server solutions *(Subversion, ClearCase, Visual SourceSafe, Starteam, Team Foundation Server, Rational Team Concert, etc.)* seem to be the more appropriate for the corporate development as opposed to the local only *(SCCS, RCS, PVCS)* and distributed *(GNU arch, Git, Bitkeeper, etc.)* products

CONCEPTS

- Branch
- Fork
- Change set
- Commit
- Delta compression
- Diff
- Merge
- Tag
- Trunk



 © Copyright 2011 Hewlett-Packard Development Company, L.P.

TOOLS, MEANS AND METHODOLOGIES

- CM should be properly organized in the very beginning of the project lifecycle
- People with the relevant expertise should define and implement configuration management methodology and instrumentation. DIY is not always a good idea
- The appropriate IT infrastructure must be established and maintained (servers, backups, networking)
- CAD database sharing might indeed require some additional investment.

 © Copyright 2011 Hewlett-Packard Development Company, L.P.



Q&A

 © Copyright 2011 Hewlett-Packard Development Company, L.P.

