



# Managing globally distributed Configuration Management

Hans Thelosen

Corporate Innovation & Technology - Design Data Management

May 23, 2008

Vrije Universiteit, Amsterdam



# Introduction

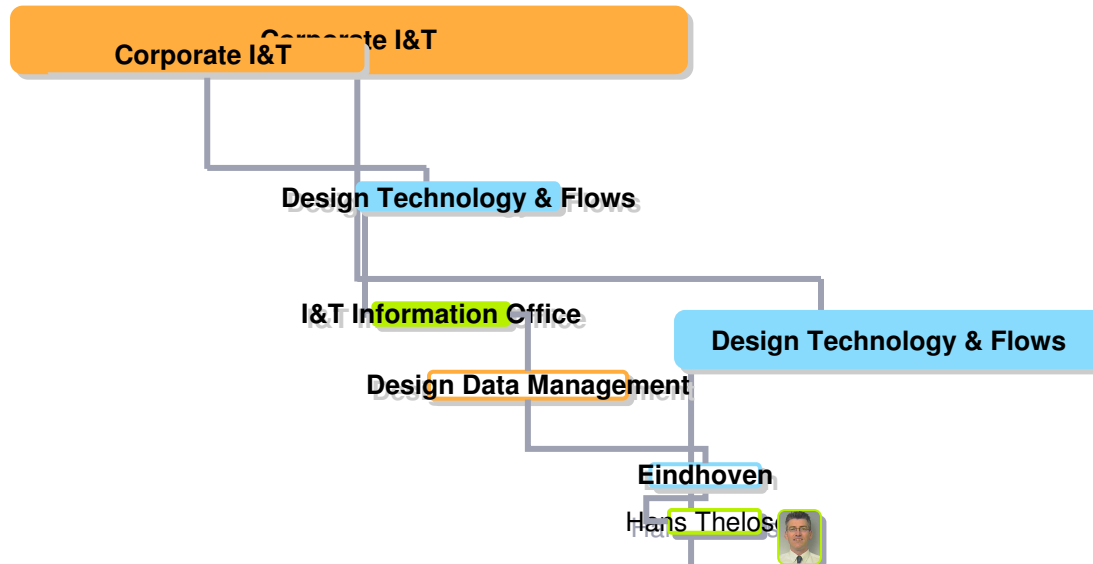
To NXP

# NXP Semiconductors

- ▶ Established in 2006 (formerly a division of Philips)
  - Approximately 37.000 employees
- ▶ Builds on a heritage of 50+ years of experience in semiconductors
- ▶ Top-10 supplier with Sales of € 6.32 Bln (2007)
- ▶ Headquarters: Eindhoven, The Netherlands
- ▶ Key focus areas (Business Units):
  - Mobile & Personal,
  - Home,
  - Automotive & Identification,
  - Multimarket Semiconductors
- ▶ Owner of NXP Software: a fully independent software solutions company
- ▶ An R&D powerhouse
  - 25,000+ patents
  - 26 R&D centers world-wide



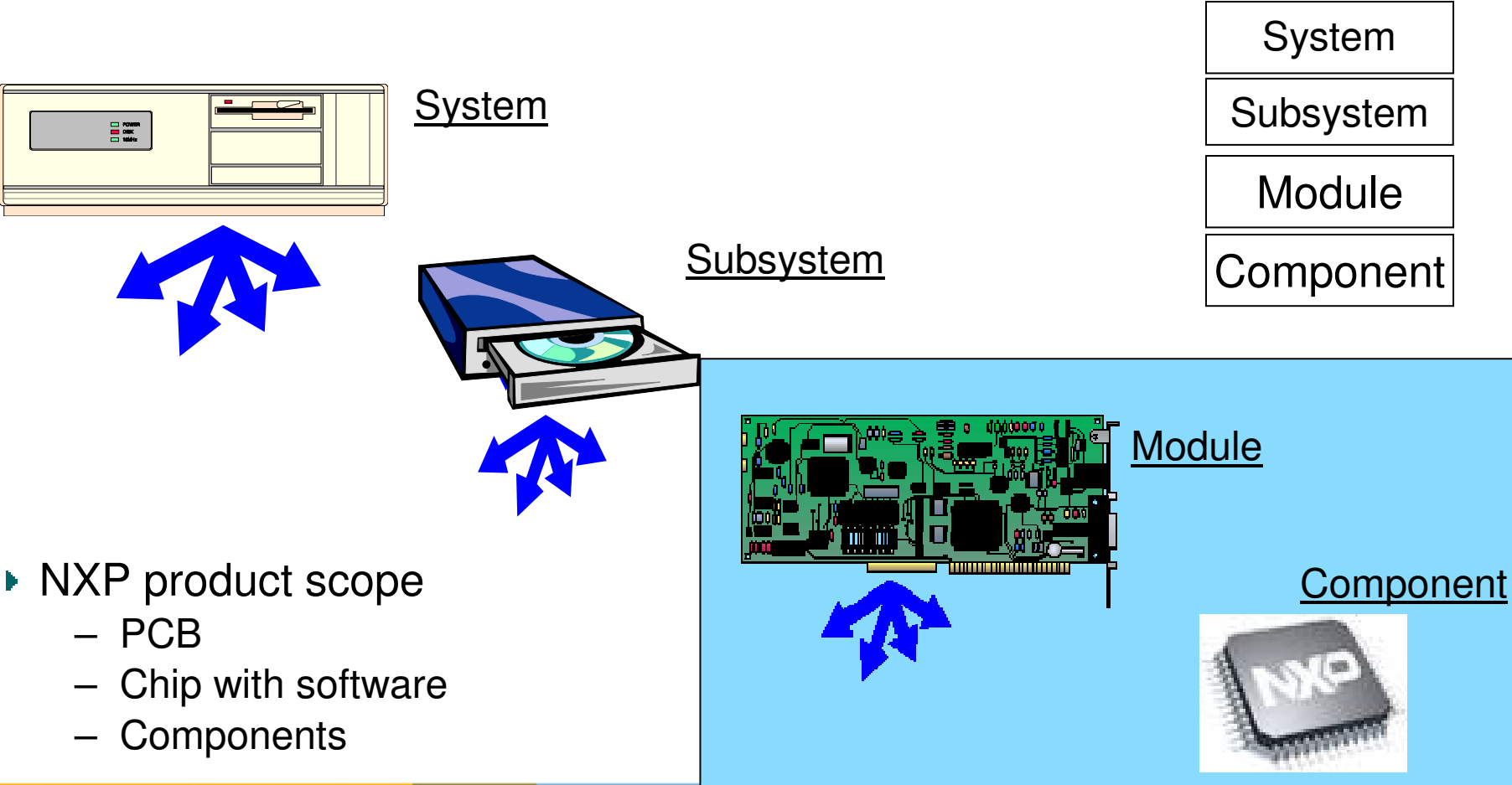
# Organization



- ▶ Member of iNTCCM
- ▶ Conducting Configuration Management training at the Stan Ackermans Institute – OOTI (Eindhoven)
- ▶ Member of the team supporting Configuration Management within NXP



# Product breakdown in the outside world



# Topics

- ▶ Managing Configuration Management
  - Follow trends in the market
  - Follow Organizational changes
    - Business Unit M&P is separated (Joint Venture).
    - Business Line Settop Boxes acquires parts of Conexant.
  - Upgrade tool versions
  - Help developers with operational activities
  
- ▶ Improvement areas
  - More efficient
  - More effective
  - State-of-the-art processes and implementation

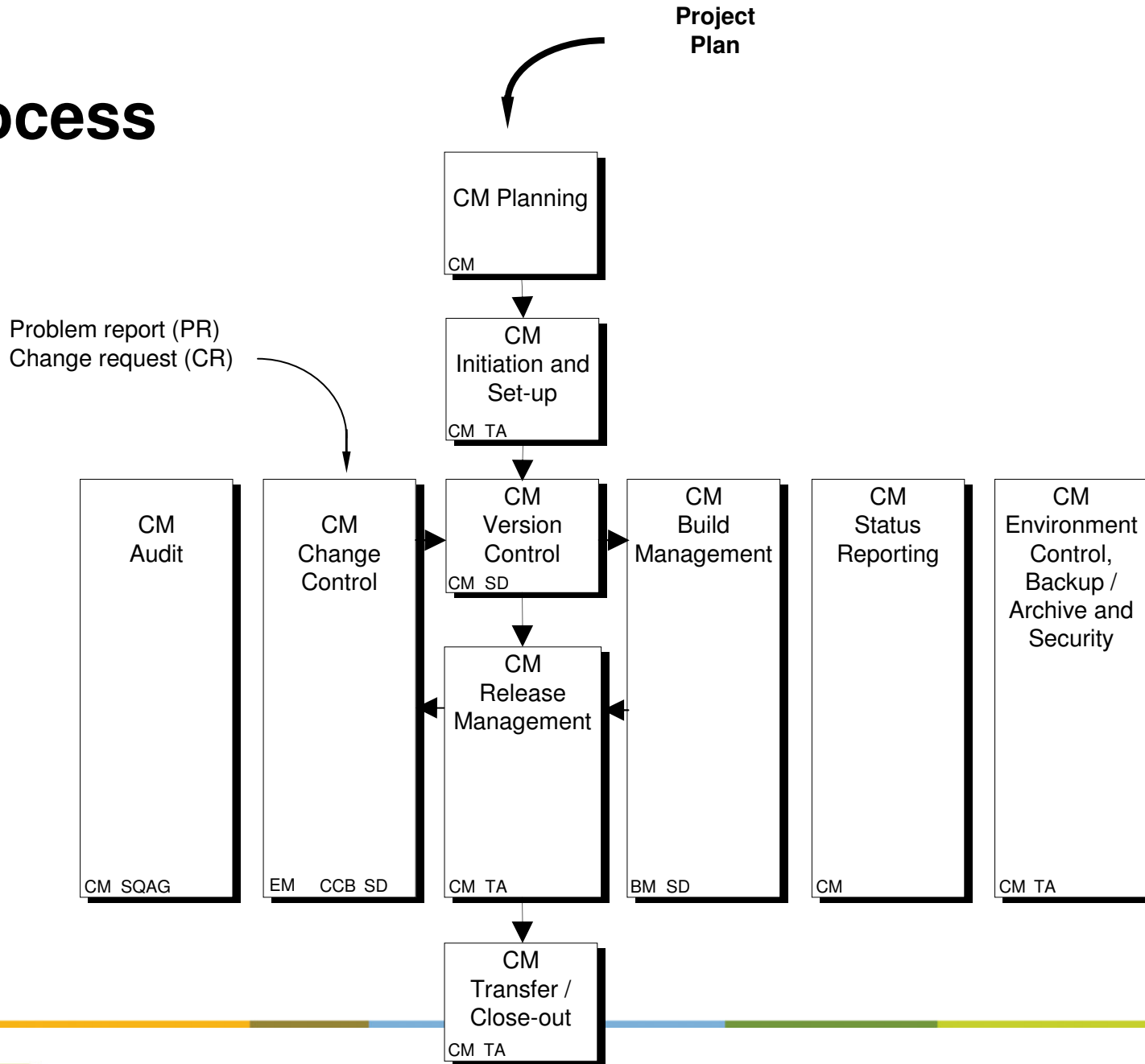
# **Operational aspects**

Support daily business

# Large / small projects

- ▶ Hardware development
  - IP (parts of a chip development)
- ▶ Software development
  - Components, programs
- ▶ Software platform development
  - Legacy
  - Re-use
  - Multi-products
- ▶ System development
  - Hardware / software interaction
  
- ▶ Almost every project has more than 1 site involved

# CM Process



# Project competence

- ▶ Most projects have their own:
  - Build management
  - Change Control Board support
  - Configuration management
  - Tool expert (depending on the site mature / expert)
  - Release manager
- ▶ If projects work in combination with other projects:
  - Alignment tools required
  - Alignment releases required
  - Alignment way-of-working required
  - Alignment promotion model (what kind of quality) required
- ▶ Internal project co-operation or with customers / suppliers

# Once upon a time there was a frog sitting in a tree.

- ▶ The frog was looking around, just trying to understand the world. According the frog there should be more than only some wetlands and some trees. He looked at a strange tree on four trunks and that tree looked at him. He tried to jump into that tree but he fell down.
- ▶ "Hello frog" said the giraffe (as the giraffe was that strange tree). "How are you today"?
- ▶ The frog said: "Wow, a tree that can talk. So I guess you are no tree"!
- ▶ The giraffe said: "No, I am a giraffe. I am here to drink some water".
- ▶ "Okay" said the frog, "So there is more than just some wetlands and trees".



# The frog

- ▶ Trying to catch bugs around him
- ▶ Able to swim, dive and jump
- ▶ Camouflage colors
- ▶ Different kind of noise
- ▶ Some live on the ground others in the trees
- ▶ Remarkable lifecycle:
  - Frogspawn
  - Tadpole
  - Froglet
  - Frog



# Frog = the build manager

- ▶ Busy with operational part of configuration management
- ▶ The build manager support the development team to 'catch/solve' bugs
- ▶ Supports an efficient team, or arranges to get one w.r.t. configuration management
- ▶ Manages several builds and the quality of these builds
- ▶ Is an expert on the tooling used
- ▶ Should be able to 'jump' to a newer tool-version without problems
- ▶ Build management requires minimal effort in a smooth environment. Only directly relevant in case of changes, e.g.:
  - Product, Technical Environment, Development approach, Development team
- ▶ Build management is not easy to learn due to the technical issues and the human interaction required (often the frog-lifecycle is somehow applicable).
- ▶ Best practice is learning on the job (from tadpole to froglet to frog) in combination with technical (tool) training.

# The build manager (1)

- ▶ Can only do his/her work properly in case the short-term and longer-term approach on configuration management is clear.
- ▶ Has to automate all daily activities where possible.
- ▶ Guide the development members from newbie to skilled on CM competence.
- ▶ Has to safeguard the operational CM competency in the development team (check out / check in / release labels, etc.).
- ▶ Should take operational responsibilities on availability, quality, performance:
  - Check consistency of the tool-repository
  - Arrange cleanup due to development crew changes
  - Arrange cleanup due to project changes
- ▶ Should be able to detect the developer with special requests:
  - multiple work areas, etc.
- ▶ Is often guided by the lead-integrator / project leader.



# The build manager (2)

- ▶ Scope is very often operational due to the pressure of the day and the direct communication with the project leader.
- ▶ Scope should be also strategic:
  - Introduce build tooling reducing the human resource workload, e.g.:
    - cruise control
    - Visual Build (introduced at some places within a Business Line)
  - Structural information providing, e.g.:
    - web-based build information (number of builds and the results of these builds)
- ▶ Build manager should be able to cope with all parts of configuration management. Not only build management and version control but also change control, release management, promotion management, audits, etc.
- ▶ The build manager should be able to understand the world around.

# The giraffe

- ▶ Having the overview on tall legs and a tall neck
- ▶ Only bending over when he wants to drink (but that is not a very safe position)
- ▶ Able to resist stormy weather (not a helicopter view)
- ▶ Able to run up to 60 km/hour
- ▶ Camouflage pattern (between trees)
- ▶ Living in small groups
- ▶ Scared of height



# Giraffe = the configuration manager

- ▶ A configuration manager shall have an overview of the strategic direction of the development project / department.
- ▶ Has to decide on, or at least guide, process issues:
  - Closed-loop change control versus Feed-forward change control
  - Closed-loop change control for Design-In projects with direct end-customer interaction
  - Feed-forward change control for product development internally
  - Decision taken by development manager but prepared by CM
  
  - Quality of releases
  - Releases contain sometimes partially solved CR/PR's.  
Due to the project pressure and customer desire, these release are created and delivered. Despite the lower quality.  
CM can arrange “tell what you have, and deliver what you tell”.

# The frog and the giraffe (1)

- ▶ To be able to implement good configuration management you need:
  - View on the future : IT, tool & process
  - Cost-down drive : if others can do it cheaper, we can do it also!
  - Quality drive : “tell what you do and do what you tell”.
  - Expertise on the tooling
  - Flexibility on business adaptations
    - Interaction with different sites, suppliers
  - Flexibility on operational hours
  - Understanding the current and future skills of the team
  - Definition of responsibilities
- Be able to find clean water and fresh air.
- ▶ The build manager arranges the main part of the operational configuration management
- ▶ The configuration manager has to create/guide the continuous CM environment.

# The frog and the giraffe (2)

- ▶ In the real development world often a number of build managers and configuration managers have to work together.
  - Some provide the strategic information on tool roadmap, development portfolio, etc.
  - Others arrange the operational aspect to support the development organisation.
- ▶ An optimum depends on, e.g.:
  - maturity of the development crew (lot of newbies versus lot of experienced)
  - flexibility of the development organisation (stable versus ad hoc)
  - number of parallel teams / products / branches
  - number of customers with different development requests
  - number of geographic location involved
  - maturity of the tooling / infrastructure
  - So: It differs almost continuously!
- ▶ BL-TV has build up experience in multi-site projects within a platform development program during several years.
  - Including development transfer from USA / Europe to India.

# The frog and the giraffe (3)

- ▶ Most of the configuration management decisions within NXP are based upon standards (high-level) and the selected tool.
- ▶ This makes the giraffe environment more difficult. The giraffes do not always speak the same language within NXP.

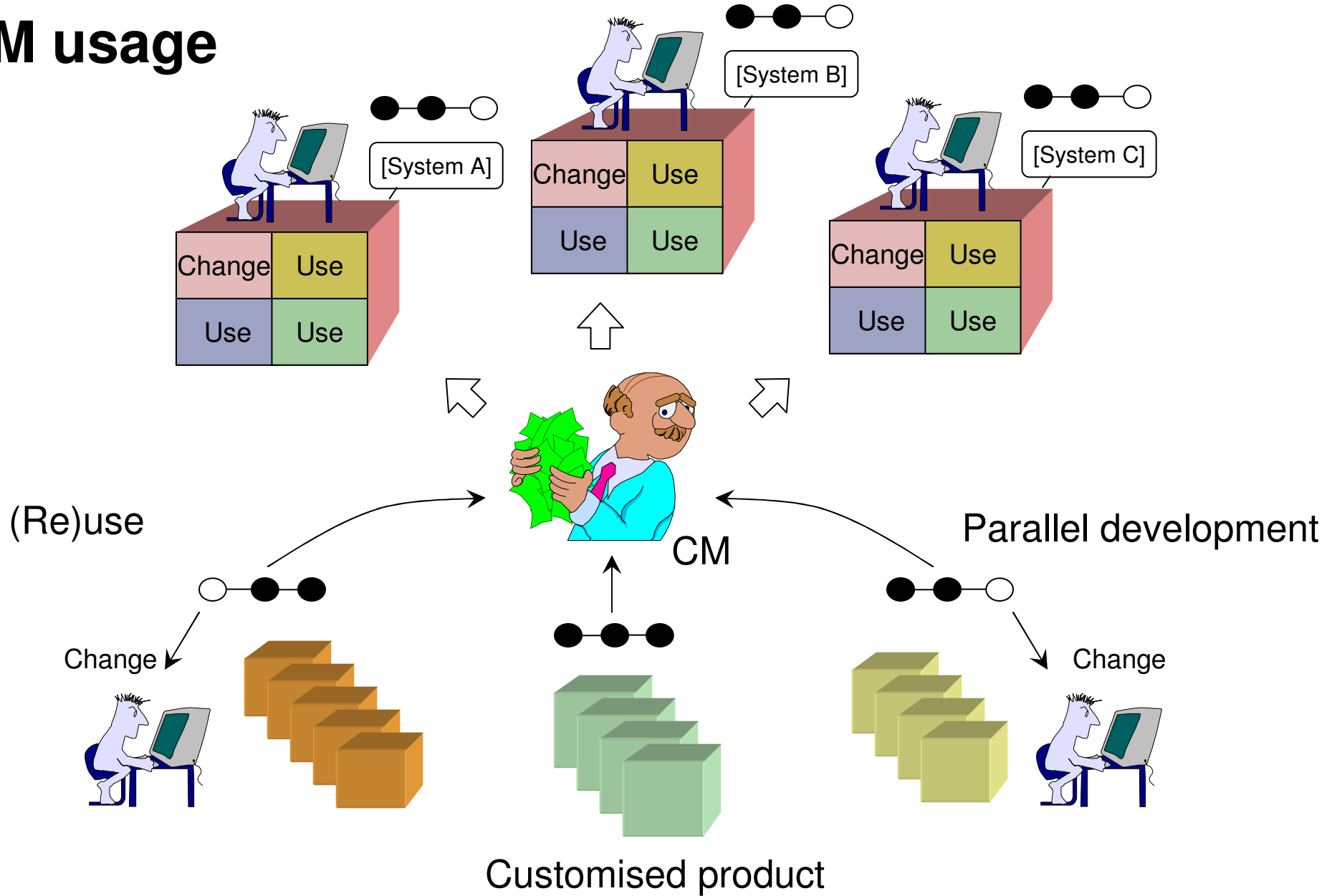
# Managing Configuration Management

Within NXP

# Configuration Management competence

- ▶ On corporate level Configuration Management support:
  - Tool supplier interaction
  - Adaptation toolset to meet NXP requirements
  - Training support toolset
  - Support adaptation to meet project specific requirements
  - Helpdesk to support knowledge exchange / get deployment feedback
  - Close interaction with operational configuration management
- ▶ Interaction on configuration management process level
- ▶ Interaction on configuration management tool level
- ▶ Interaction on configuration management infra-structure level

# CM usage

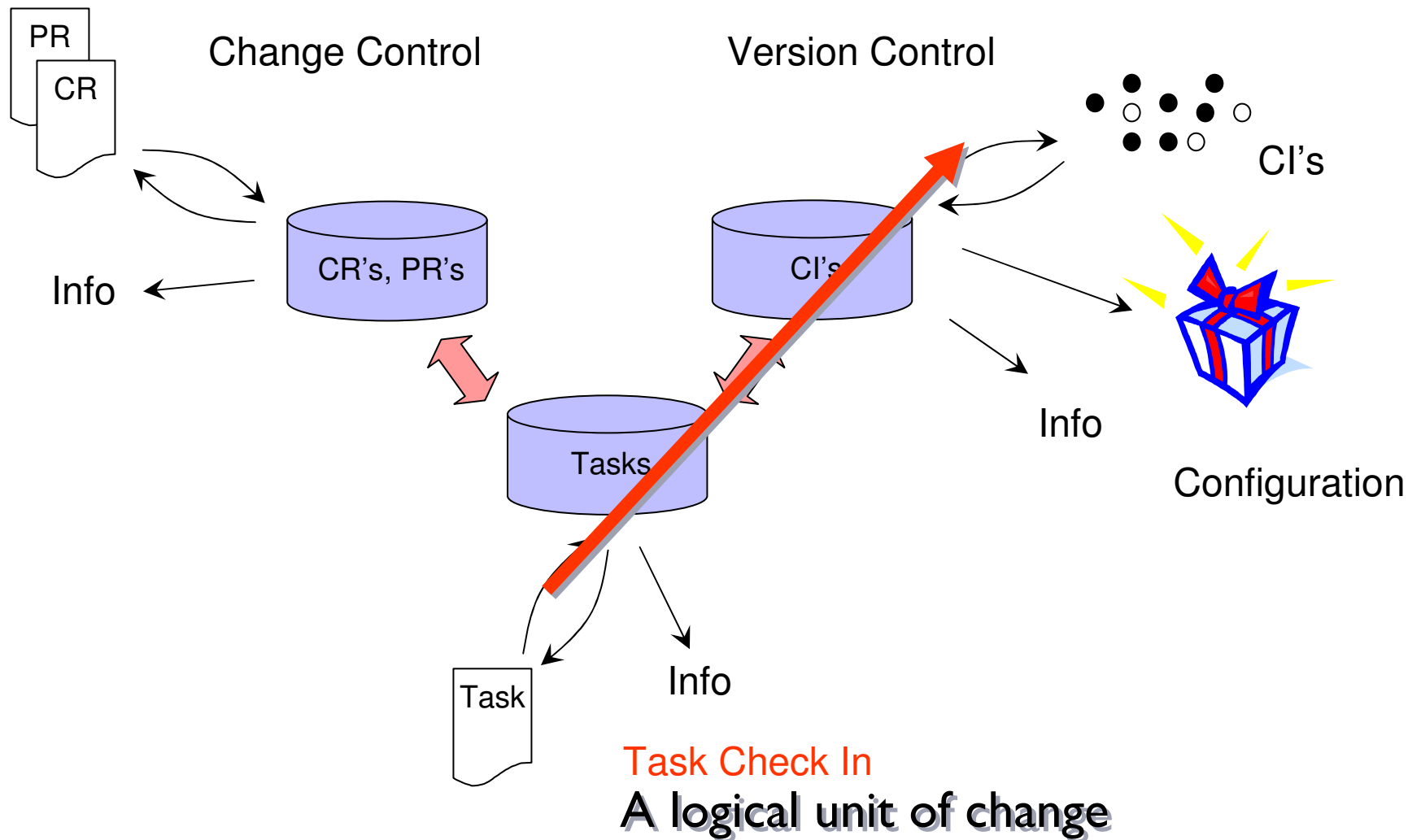


# Global configuration management support

- ▶ Enable collaboration by means of tools
  - Collaboration tools/providers: WANdisco, CollabNet, GForge
  - Multi-site support of tools
    - Distributed Configuration Management
- ▶ Promote, facilitate communication between projects, sites, configuration managers
  - Arrange CM workshops
  - Wiki pages (internal NXP)
  - Promote standardization
- ▶ Sharing Best Practices
  - Higher quality
    - Better control of release content
    - Less error prone development
  - Better Time to Market
    - Predictable integration flow
    - Predefined metrics



# Best Practice: e.g. Task Level Commit

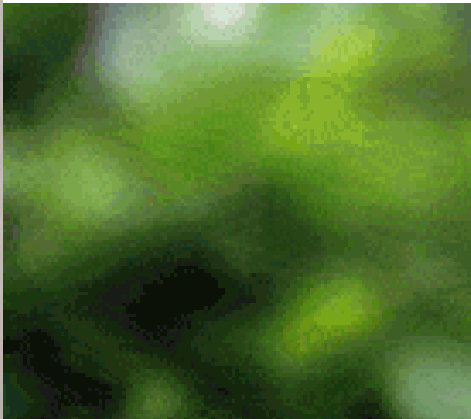


# Conclusion

- ▶ Organization is dynamic
  - Acquisition / mergers do really have impact on development
- ▶ Configuration Management tools are improving
  - State-of-the-art tooling is required by development
  - Open source related activities do have impact
- ▶ Multi-site collaboration
  - Is almost a standard requirement
  - Not all knowledge / skills can be expected at all sites
- ▶ A single silver bullet, doesn't help!
- ▶ Don't arrange too many details as the flexibility to change will always be required.

# If you look long enough

- ▶ You will very often see somewhere a giraffe between the frogs ;)



- ▶ Some frogs look like giraffes and vice versa ;)
- ▶ And no, the giraffe didn't kiss the frog.....

