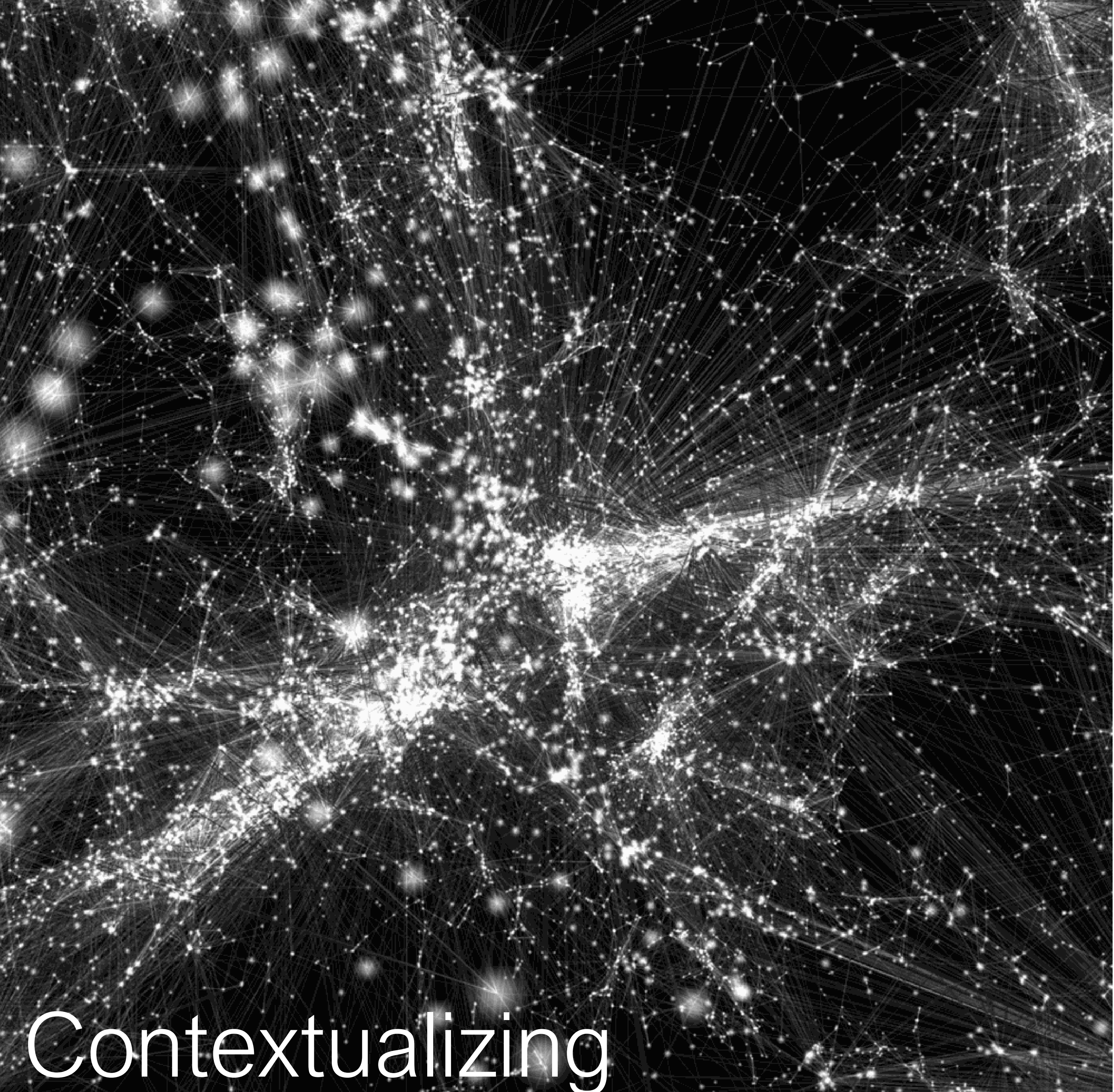


Contextualizing Configuration Management

Conclusions from CM FORUM workshops 2015-2016

The importance of context

Each CM FORUM workshop during 2015-2016 has presented a new CM perspective. Not only do the host organizations have very different products and are focused on different phases in the lifecycle, they have also chosen different ways to assign responsibility and mandate regarding CM.

Every workshop theme has therefore been placed in a new context, dependent on system, lifecycle phase and organizational approach to CM. A diversity reflected in this paper, which seven chapters offer different ways of contextualizing CM.

With the increasing complexities of technologies, services, networks and interfacing organizations, the configuration manager must find ways to make day-to-day CM activities everybody's responsibility. To do this, all co-workers must not only be convinced about the overall importance of CM, they must also experience that the effort they put in add value to the end purpose.

Since the effect of CM is so dependent on the awareness, will and competence of all co-workers, the configuration manager's challenge is to identify how methods, rules and routines can be applied and best supported by available IT tools, to make CM function as smooth as possible. In order to make the right strategic decisions, the configuration manager must understand context, and take into account:

- *the system architecture and the system boundaries for the configuration that is within the scope of the organization or project (and which stakeholder that take responsibility for what is outside the scope);*
- *the system lifecycle and which phases that are within the scope of the organization or project (and which stakeholder that take responsibility for what is outside the scope);*
- *the applied development model and how it is best supported in terms of baselines, status accounting and configuration control.*

Taking these contextual aspects are into account, CM activities can be defined, scaled and distributed throughout the organization. If successfully implemented, the role of the configuration manager can be oriented towards upholding co-workers CM competence, monitoring and continuous development of the CM capability.

Tobias Ljungkvist, editor

Thank you for contributing with text and figures, review comments and general feedback:

Per Kapfält, Yngve Ahlenbäck, Ulf Carlsson, Mats Ireby, Margaretha Eriksson, Peter Bártfai, Staffan Westbeck, Ulf Asklund, Johan Ranby, Helena Laitila, Tom Strandberg, Gabriel Ordeus, Leona Pirhi, Marina Omer

The content in this paper is the documented output from the following workshop events:

Can CM find common ground for HW and SW development?
Syntell, Stockholm 2015-03-05

How can CM enable business downstream?
Saab Surveillance, Järfälla 2015-04-23

How can CM support Requirement Management in a product line environment?
Volvo CE, Eskilstuna 2015-09-22

CM challenges in a networked society
Ericsson, Kista 2015-12-02

Organizing CM to support the program, project and line relationships
Nordea, Stockholm 2016-05-03

Strategies to establish a stronger mandate for CM
Bombardier, Stockholm 2016-09-15

Mastering the art of coordinating configuration ownership
Vattenfall Ringhals, Våröbacka 2016-10-20

Taking CM into the networked society

How do we develop CM to meet the challenges of developing, managing and operating products in a society where technology is increasingly interconnected?

Which dependencies exist and which type of relationships are necessary to establish? Are the circumstances to achieve traceability better or worse?

In the networked society a flexible and efficient infrastructure enables a large amount of products, services and users to be connected. This resulting in millions of new use cases; new possibilities to customize and upgrade products and services during operation and new business models. For complex products the possibility to create long lasting relationships with customers through innovative service offerings tend to give much more revenue than selling the product. These relationships involve an increased amount of traceable data about customer activities compared to a traditional sell-buy relationship. Commercial actors have various opportunities to adopt their businesses to this development:

- ThyssenKrupp Elevator, one of the world's leading elevator manufacturers, maintains more than 1.1 million elevators worldwide. Drawing on the potential of the internet of things by connecting its elevators to the cloud, gathering data from its sensors and systems and transforming that data into valuable business intelligence, ThyssenKrupp is vastly improving operations. They can now go beyond the industry standard of preventative maintenance by offering predictions for preemptive maintenance, resulting in possibilities to guarantee a higher uptime percentage on their elevators. They now see the possibilities to offer service contracts for elevators from other brands[1].

- The travel industry where for example TUI through their own applications and a complex network of suppliers together are delivering "experiences" in a dynamic world where offerings are constantly updated. The whole environment need to be online year around and the results of a complex mix of activities need to be incorporated on the fly including updates to the applications, planning for future seasons, contract negotiations with suppliers, presentation of offerings, handle reservations, customers actual travelling, management of delivery problems and complaints, feedback to the Marketing Department, etc.

- Relations are established between product instances. For example, relations between two cars can enable services such as slippery roads warnings or relations between two chainsaws makes sure that they alert the users if they are in too close proximity to each other.

- Services provided by music distributors such as Spotify are customized and updated continuously. Data related to customer profiles (behavior, health, driving) can be used to tailor offers.

Given the description of what signifies the networked society above, several implications on Configuration Management can be highlighted:

Possibilities for new services increase the need for control

The large increase of services poses new traceability requirements; such as what product individuals are connected to each other, or which services are currently delivered by a particular product instance. Many of these services might not be new for the producing organization or customer, but the necessity to keep control over these services is increased since more dependencies are introduced in the network. Examples of factors adding to the complexity are:

- New variants of a product are introduced due to several subscriptions are offered the customer (12 months, 18 months, etc.)
- 24h support is offered for the product
- Services are offered as pay-for-use

An important challenge is therefore to ensure that CM practices are spread to the service domain.

The system designer and the end user must be brought closer together

CM needs to facilitate the possibility to predict changes in customer demands and requirements. It must be possible to baseline requirements from all phases in the lifecycle to enable traceability between requirements and design.

Possibilities facilitate operational feedback (failure reporting) sets new demands on the capability to react and respond to the feedback. E.g. bugs in open source software, the supplier must know if they use the particular software. Ideally, the network enables for the supplier to "ask" the product individuals if they contain the particular software.

Increased need for control of product individuals

In a networked society there is an increased need and an increased possibility to keep track of the installed base (which software is installed in which product individuals). Preferably, a product individual "knows" its own configuration.

Large diversity of interconnected data to consider

Suppliers generate data in different formats, causing needs to adapt information to make it accessible and understandable for different stakeholders. CM needs to access the information in a standardized way (e.g. product identity, attributes, etc.).

Increased number of information interfaces drives the need for standardized formats for the information. Note that applicable standards are also a part of the configuration.

The network architecture that the product or service depend on requires configuration management

For all products that are part of networks, the following builds the architecture for the network that the product is connected to and therefore will need to be configuration managed:

[1]See: http://blogs.microsoft.com/firehose/2014/07/16/the-internet-of-things-gives-the-worlds-cities-a-major-lift/

- APIs
- Protocols
- Transmission capability
- Tracing connectivity path
- Log

Products might need to be able to run several versions of the architecture. An instance of the product will need to be able to shift between versions, since other instances in the network will be upgraded at different points in time.

Needs to expand the test environment to include the network context

In a networked context the production environment need to be updated more or less continuously. Examples are:

- Enhanced functionality for end users
- Enhanced business management capabilities
- New nodes are introduced
- Changes in network topology and capacity
- Support of new communication protocols
- Added computing or storage capacity
- Bug fixing in network or applications

To secure uninterrupted production, four types of testing need to be streamlined:

- Development environment testing by team of developers where the end result is an integrated environment that is handed over to next stage.
- Testing environment separated from the development environment.
- Acceptance test environment where the readiness to be handed over to production is evaluated.
- Continuous monitoring of performance and events in production environment.

One major challenge in a networked context is the fact that events in nodes of the system are asynchronous. This means that things can work as expected in a development environment for a part of the system but in the full production environment, capacity related conditions could result in failures in parts of the system if not handled correctly.

New advantages for CM that come with the networked society

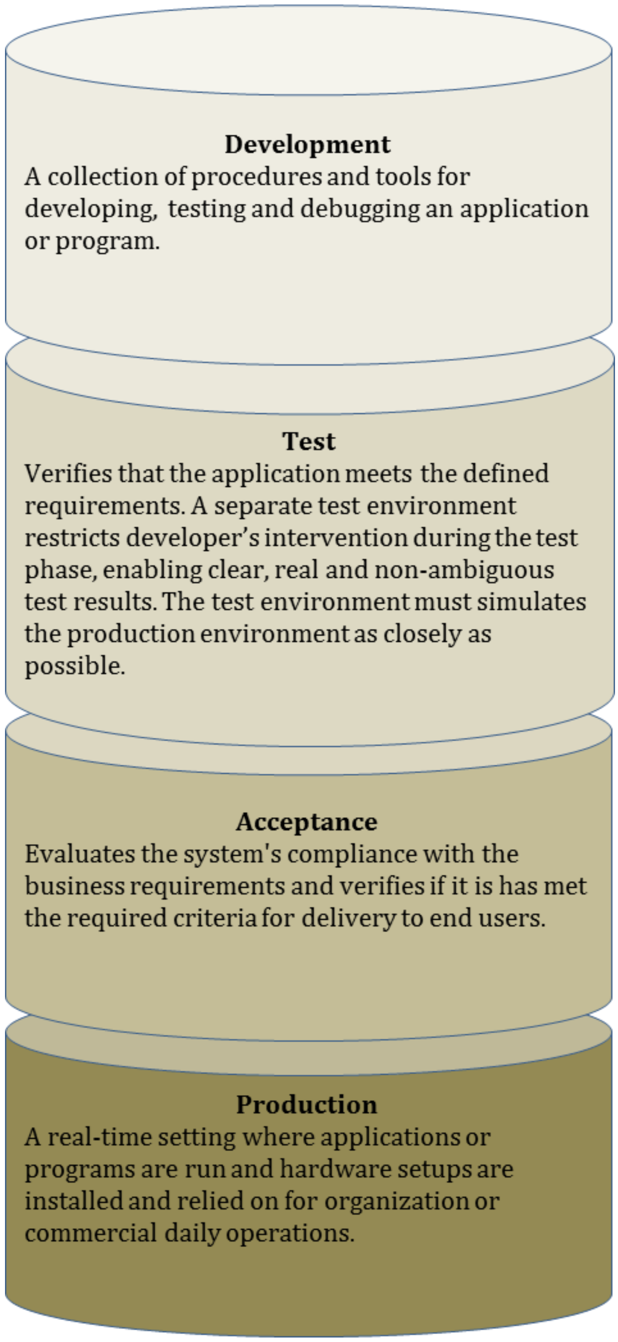
Easier implementation of changes

For software changes, the step from deciding that a change is to be introduced to the finalized implementation can be automated.

Status accounting is better supported

Several aspects of status accounting are better supported and in turn provide better basis for quality control, addressing and notifying changes, design decision and business ventures:

- Access availability, data can be assembled and presented on demand, for example: an individual Bill of Material can be maintained in the cloud.
- Data can be visualized graphically in much more sophisticated ways.
- Better support for data integrity (access rights, logs etc.).
- Putting services in the cloud gives high flexibility in selecting processing and storage capability for special events and increasing customer load.



Note that this support is not established by itself, it has to be developed. Also, many times it requires that the customer agrees to give access to personal data.

Diagnostics is better supported

The network makes it possible to connect product instances installed at remote sites and monitor performance, run diagnostic tools, change configuration, update software. System status and parameters can be logged live and traceability throughout the complete chain it gives possibility to receive information back from anywhere and pursue performance based diagnostics.

Centralized tracking of product/services performance in real time results in possibilities to minimize downtime, schedule preventive maintenance as well as detect need for maintenance.

Possibility for customer/user to perform configuration auditing

The possibility to verify if you have the latest/correct software is a way of auditing the configuration of your product.

The configuration manager can work anywhere

Web and cloud solutions provide flexibility in terms of administrating product configuration information.

The user more likely to "trust CM"

Increased frequency of releases of software has led to customer acceptance that software version is not labeled on the hardware (the version does not need to trigger new version up to system level).

CM processes and methods that need to be developed

Overall there is no need to reinvent CM methodology. Current methods will be useful but they need to be applied to larger quantities of objects and data. The role of CM need to be to ensure that adequate processes and functions are implemented in new phases/organizational entities/ across new interfaces etc.

An example of this development where CM finds new ground is DevOps that for instance enables automatic update of total system. Another example is how organizations are moving into a more agile approach thus narrowing the gap between engineering phases that traditionally have had less interaction.

Examples of CM-related methods that must be further developed are:

- Methods need to be developed to ensure that CM is applied in later stages in the product lifecycle to a greater extend (not just from development to production)
- The lifecycle model for the service need to be formulated, and probably services need a different set of metadata.
- Services now need to be regarded as CI:s so that relations between products, customers and services can be established, enabling service-oriented views of the system
- It will be necessary to define the "levels" that CM is to be pursued in with agreed methods and processes. Levels not only having to do with a particular system hierarchy, they could also be constrained to a defined network (with limited capacity), a given purpose (such as a customer campaign) or application of virtual servers.
- Configurators will enable end user product specification, and to be used when order/buying products, or to enable billing. There is a risk with configurators. It is difficult to verify accuracy, the quality of the input, understanding the implemented rules.

CM in a lifecycle context: *the Concept phase*

Consider a supplier of wheel loaders. With current generation wheel loaders still operating at several customer sites, a project is initiated to develop a new generation wheel loaders. Five different sizes are defined (6 tons, 10 tons, 15 tons, 20 tons, 30 tons). Each size will have 5-10 variants (e.g. different engines, different tire sizes, different loaders etc.).

In what way should the requirements be structured?

What should be considered from a CM point of view regarding updating current requirements and adding new requirements?

For CM it is important that requirements are defined based on architecture. Based on the architecture, generic requirements can be identified and constituted as the generic "platform" that enabled for reuse of requirements data. Refer to the figure below. After the generic requirements are defined, the delta for each variant can be structured.

The following considerations regarding updating current requirements and adding new requirements are important:

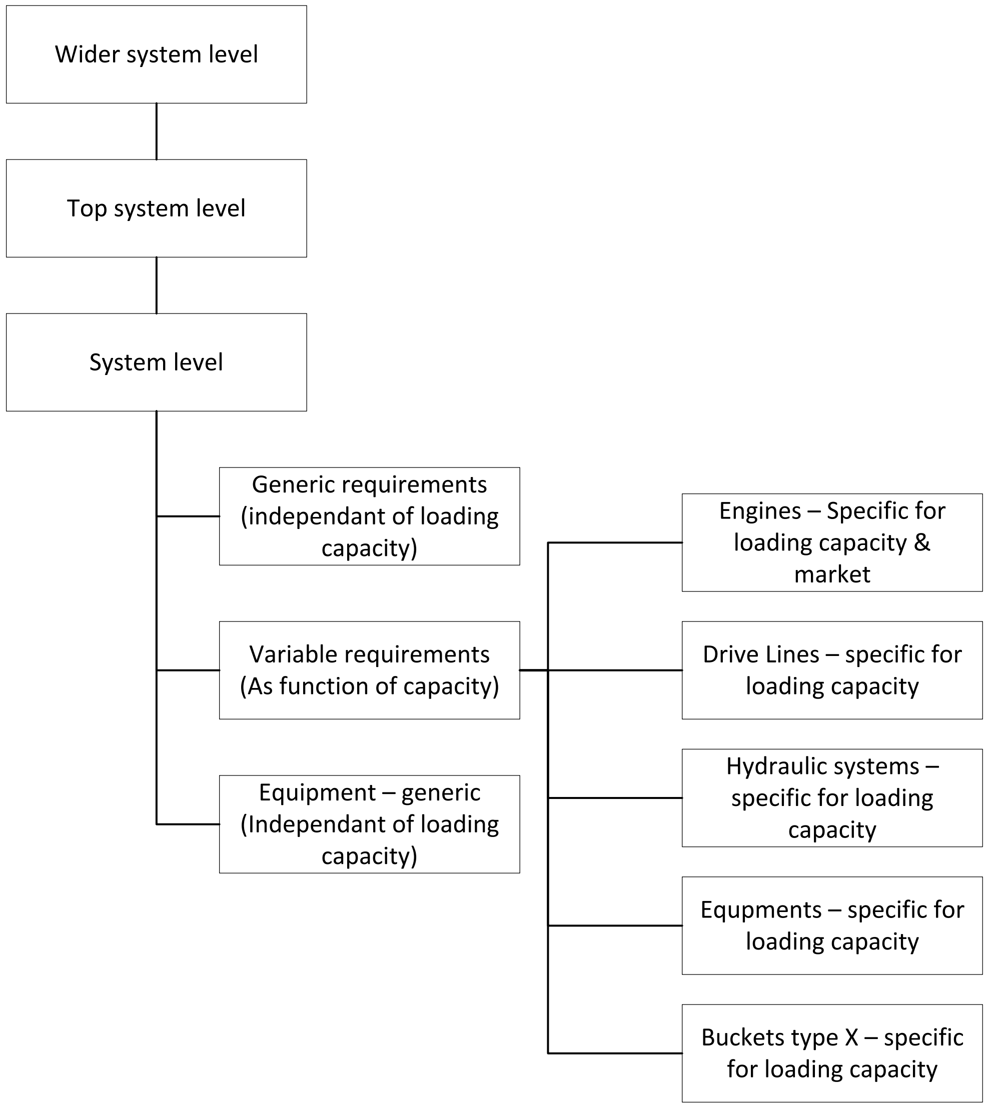
- As soon as formulated requirements have reached a status where they are used as input to design activities, they should be baselined and then formally controlled by means of a change process. For design decisions, the information that the decisions are based on also needs to be recorded.

- It is necessary that a relation is established between a requirement and the system it affects. When a design change is introduced that affects a requirement for a particular system, a new version of the requirement must be formulated and related to the new version of the system. The old requirement is valid for as long as there exist older versions of the system.

- If new requirements are introduced, it is important to identify what existing requirements are impacted / obsolete due to the new requirements (impact analysis) to maintain consistency with the complete requirement set.

- As part of status accounting of requirements, there should be a record of when a requirement is no longer valid. Otherwise, uncertainty if a requirement is valid or not is introduced.

- If requirements are to be updated, it is important to consider the approved system of interest (or "design space" i.e. the scope in which changes are allowed to occur, usually defined by form-, fit-, function, performance of function and characteristics of functions). CM should be applied to enable evaluation of design changes given the defined system of interest and to make sure that if the system of interest is transcended, adequate design authorities are involved.



CM in a lifecycle context: *the Development phase*

Configuration Management practices must be adapted to the chosen development model: Waterfall, Vee, Spiral, Incremental, Agile etc. But for complex systems that integrate both hardware and software, sticking to one single model might not be possible.

Instead, Configuration Management will have to be adapted to manage different sets of lifecycle stages, different needs for iterations and different approaches to baseline planning and release management.

The differences between CM practiced in an agile development environment compare to a V-model development environment are, for instance:

- Agile methods require more CM activities at the start and end of the development process and more focused on interfaces than within the development process, whereas traditional development requires more CM activities during the development process.
- The delivery from a sprint need to be baselined and change controlled, but formal change control is not pursued within the sprint.
- The method of documenting the system requirements differs between traditional development and Agile. Agile documents stories instead of specifications.
- Different terminology in various development models.
- Different delivery steps.
- Different life-cycle perspective, how can one compare the results over time.

To find common ground between development models, in particular for integrating HW and SW development, the following is important to consider:

- Communicate the "complete picture", the goals and common milestones (synchronization points). E.g. launch of a virtual product. Provide baselines to all development groups.
- Goals and principles shall be common ground. Methods and tools may be different, sometimes by need, sometimes by legacy (difficult to change). HW and SW teams work in different contexts with different needs.

CM in a lifecycle context: *the In-operation phase*

CM in an ITIL organisation

The IT Infrastructure Library (ITIL) best practices standards include specifications for configuration management. According to ITIL specifications, the four major tasks of configuration management are[2]:

- Identification of configuration items to be included in the CMDB
- Control of data to ensure that it can only be changed by authorized individuals
- Status maintenance, which involves ensuring that current status of any CI is consistently recorded and kept updated
- Verification, through audits and reviews of the data to ensure that it is accurate.

ITIL CMDB

A Configuration Management Database (CMDB) is a database that contains all relevant information about the components of the information system used in an organization's IT services and the relationships between those components. A CMDB provides an organized view of data and a means of examining that data from any desired perspective. Within this context, components of an information system are referred to as configuration items (CI). A CI can be any conceivable IT component, including software, hardware, documentation, and personnel, as well as any combination of them. The processes of configuration management seek to specify, control, and track configuration items and any changes made to them in a comprehensive and systematic fashion. As, such a CMDB is similar as other Product Data Management (PDM) databases but with adaptions to suit/fit the ITIL framework.

The challenge is to proactively plan for defining CIs so that the input not only is reactive. When we know which CIs we have, it is possible to analyse the system itself. For instance, if CPUs are defined as CIs, they will be included in the CMDB and this makes support planning for CPU-usage possible.

The most natural structure to build in the CMDB is the one that cover the "current" status of the IT infrastructure. This can be called "as-is", "in-operation", "as-deployed" or some similar term. This is of course the most important structure - the one used for navigating the current IT structure, in particular when addressing outage or fault situations. But it is possible to get more value from your CM investment by expanding this to cover other situations. For example, to plan for a future situation where parts of the IT equipment are replaced. This can and should be planned in a CM structure that to a large degree is the same as the current "as-is" structure. But of course parts of the equipment are replaced, expanded or reduced as demanded by upcoming requirements. In this "as-planned" structure, the revised relations and workload can be analyzed to see if they fulfill the future demands. Naturally, several different "as-planned" structures can be maintained, for example to explore different options.

Another situation can be to look backwards in time, e.g. for legal or contractual reasons. For example, to answer the question "when was this equipment introduced into the IT infrastructure?". This can be either by snapshotting the "as-is" structure at relevant points in time (essentially creating

ITIL at Nordea

At Nordea, an automated process has been developed to import the data to the CMDB that includes validation of data quality. The added values are:

- Improved impact models based on relations.
- Automated location and placement data for all operated Hardware.
- An expanded location model for other asset data.
- Improved location data quality (reliable data).
- Improved process efficiency, less time spent.
- Verification can be performed.
- Improved traceability.

[2]See: http://whitepaper.talentum.com/whitepaper/view.do?id=18079 for further reading.

historical baselines) or by constructing the CMDB to handle history and enable recreation of random-time historical situations.

Such a design of the CMDB necessitates the introduction of a lifecycle concept for all CI. This makes them visible in the CMDB with indications of "in operation", "planned", "retired" and similar status, with corresponding status change log. In addition to the CI, all relations should also be given life cycle status, so that there is a flexibility in the configuration of the IT infrastructure.

General features of the CMDB, preferably includes:

- Where it could facilitate, set restrictions and mandatory fields for data input.
- Automated support to compare data (run frequently)
- Validate check of inputs and drop-down lists when free text is not an option

When starting up a project, it is important to be involved in planning deliveries and structure development work. The configuration manager should be involved from the start, see where the project “goes” and be able to set early requirements for CM.

It is important to have an established CM process in place, that covers the entire lifecycle. The process should be well described and easily accessed and easy to follow, preferably including support/helpdesk. There must be governance for the process. And routines for follow-up on CM activities and processes from line management - to be a natural part of project lifecycle, including a corrective action process. The process need to be aligned with:

- FMEA (Failure Mode Effect Analysis) and quality control.
- Architects to improve CM in the organisation since architecture and CM structure goes hand in hand.

Audits of the CMDB

All co-workers should have the responsibility to report incorrectness. But planning audits is a responsibility of the configuration manager on behalf of quality assurance.

The most important thing is to make sure that the data is correct at the start, since it is very hard to make significant data quality improvements with audits afterwards. How often this should be done depends on how critical the system is. For systems under development audits should be done at toll gates/milestones. This should be the responsibility of the system owner, the configuration manager and system architect or other specialists with knowledge enough to be able to find inconsistencies and bad data.

An audit program can include internal audits initiated by the configuration owner, that can invite needed attendants and performed regularly (for instance twice a year) as well as regulatory required audits (for instance yearly).

CM in a lifecycle context: *Across phases*

What requirements on the CM function in the early lifecycle phase are necessary to exploit business opportunities during utilization/support and retirement lifecycle phases?

What role can CM play in order to facilitate addressing these opportunities?

Practical advice to accomplish CM for the whole lifecycle, include:

- Define service concept for the product. Important to scale against what the organization is capable of in terms of integrated logistic support.
- Formal change management need to be implemented "all the way" to product individuals.
- Customer feedback loops must be required and controlled during and after guarantee-period.
- Technical documentation and training material need to be part of change management.
- A general standard for identifying and naming structures to avoid misunderstandings (product-id, doc-id, versions, etc.).
- Suppliers need to consider what type of CM that is required from the customer to be able to give quality support during operation/maintenance phase. If this can be defined, the supplier can sell it as a service.

The table on next page typical business opportunities that, if addressed in early lifecycle phases, can create business value downstream.

Scenario: In-operation engine accident

Consider a scenario where five years into operation, an engine on one of the new generation wheel loaders breaks down. The cause is unknown, but a design fault in the engine is suspected (could be HW or SW). The specific engine variant is installed in several products and variants. What is required from a CM point of view to be able to identify all the individuals with the particular engine variant and what is required from a CM point of view to minimize the risk that this happens again?

The central aspect is to establish baselines and a change process that enables to trace the design, including HW and

SW, all the way back to early lifecycle phases, such as conceptual and development phases, because these are the phases in which requirements are formulated. This requires a structure that is broken down to a low enough level for fault-finding and baselines established at relevant points in the lifecycle (as-required, as-designed, as-delivered, etc.). This could be established by means of a Product Lifecycle Management (PLM) system with traceability from the faulty engine type and version to all vehicle types and variants it has been included in, and traceability of the faulty component version to all engine types it has been included in.

The CM responsibility is to manage large amount of data in a systematic and structured way, making information available for key stakeholders, in a comprehensive way – supporting analysis and decision making. Traceability for individuals are necessary since this makes it possible to correct, verify and validate all affected individuals. Physical labelling connected to the structures and established baselines also for each individual is important.

To connect system data (e.g. engine variant and SW-version) through engine unique serial number connected to product/ machine serial number as part of assembly sequence (system architecture, connected to assembly sequence through systems serial numbers / SW-versions). The Product/Machine serial number, shall contain a connection to machine configuration as part of assembly sequence. This shall be a delivery from the assembly line to product maintenance, after market and sales support (for connection to customer).

This enables to track changes for each individual during the operation phase. System configuration changes have to be systematically retrieved and recorded/processed by after market / service function. For the data feedback a customer register where all individuals are matched towards customers is necessary. The CM function must also be able to relate valid configurations to extract of vital reliability and claims data history from Failure Reporting And Corrective Actions System (FRACAS), for preventive actions.

A prerequisite to track changes during the operational phase is that the supplier and acquirer have contracted which party that records and share which type of information. It must be agreed which party that documents the service actions, the configuration changes that has been implemented, etc.

Usually it is not enough only to establish the traceability regarding the system itself, but also the enabling systems, such as the test environment (to be able to analyze root cause). The CM function should also be able to point out which test-cases and verification strategy that was used, so adequate new test-cases can be identified.

Business opportunity downstream	**How to address this opportunity during early lifecycle phases**	**How CM facilitate addressing this opportunity**
Selling breakdown structures, such as As-realized, As-build, Maintenance structures, Individual structures.	Well defined architecture, well integrated workflows, require feedback from customer.	Baseline support including quality assurance of baselines, Identification of CIs and data, establishing traceability between structures, design customer feedback loop.
Selling customer training.	Well-developed Logistic Engineering/ Logistic support, require information concerning the customer's maintenance organization.	Baseline support to for Logistic Engineering/ Logistic support, Identification of CIs and data for the support system.
Offering continuos maintenance support (technical publications, training etc.).	Logistic support analysis as part of the design-process, well defined maintenance structure, modularized technical documentation & training mapped against the maintenance structure.	CI identification, mapping structures, Securing that Logistic Support Analysis, Technical documentation & training involved in change management process.
Offer to retire, reuse or recycle products.	Well defined architecture that enables "modules" to be reused.	CI identification, formal change management, traceability all the way from requirements to the individual.
Full service agreement.	All the aspects mentioned above need to be addressed.	Supplier's CM must define requirements for customer's CM, alternatively the supplier's CM extends to the customer side (status accounting, change management, baseline establishment).

CM in an organizational context: *Configuration ownership*

All probably agree that configuration control is impossible unless there exists a clear ownership with authority to approve a design change or a configuration baseline. Many organizations struggle to define a scope of ownership that makes sense both in relation to system views (subsystems, functions, product lines etc.), and in relation to the way the organization itself is constituted (departments, disciplines, projects, etc.).

Another struggle related to ownership is that, in reality, ownership tends to be more or less actively exercised depending on the knowledge and experience of the particular person or group that have been assigned the ownership. So, when we for instance find ourselves juggling multiple configuration changes against the same baseline, how do we master the art of coordinating the configuration ownership[3]?

Typically, configuration ownership is assigned to the organization or person authorized to approve:

- A configuration change to a product,
- Changes to product definition information and other related documents,
- A baselines or a release (or cancellation) of a configuration and its product configuration information.

The ownership scope is either defined against a particular system/subsystem or defined as owning, or being responsible for, a particular aspect or discipline that is applied when developing or maintaining the system, such as maintenance, safety, verification etc. (referred to as "Disciplinary ownership"). Either way, the ownership is often assigned to a line organization, that also controls the necessary resources to exercise the ownership.

When a project is established, configuration ownership can be delegated to the project. The project's configuration ownership is usually then constrained to particular subsystems for a limited part of the lifecycle. Disciplinary ownership can either be delegated to the project together with the appropriate resources, or it can stay in the line in which case the line have deliverables to the project.

A challenge is to coordinate the situations when different ownerships interfere. It is important to agree who has the coordination responsibility – the line or the project.

When managing multiple changes that affect the same subsystem, and several projects, the following is important:

Stakeholder analysis

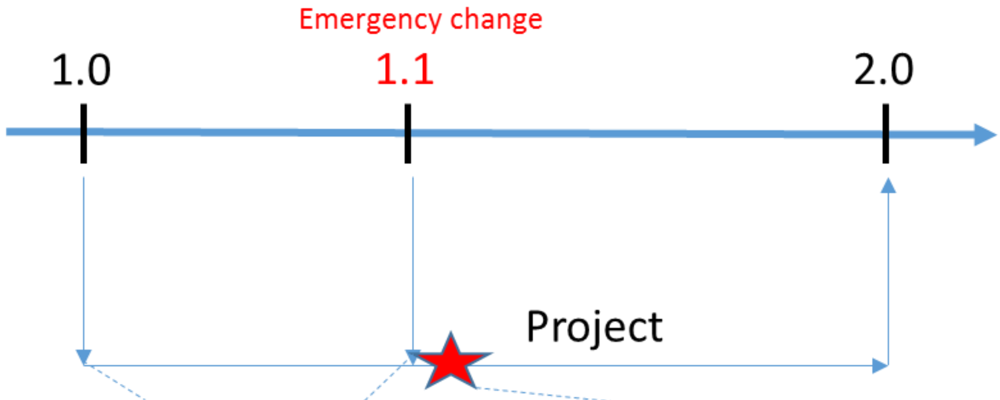
Map affected subsystems, looking at various views of the system (functional, physical, spatial, electricity, etc.) and against a time axis, to identify "hot spots" for the different stakeholders. The analysis should be updated as new versions of the input baselines are created. The stakeholder map is typically controlled on program level (if such exist).

Defining rules for establishing and updating baselines in concurrent projects

Consider the figure series on the next two pages, outlining the principles for rebasing projects against the main development track.

[3]The following terms can be regarded as aliases: CI-Owner, Approval authority, change approval authority, configuration change management authority, change control board chairperson, decision authority.

Example of rules for establishing and updating baselines in concurrent projects

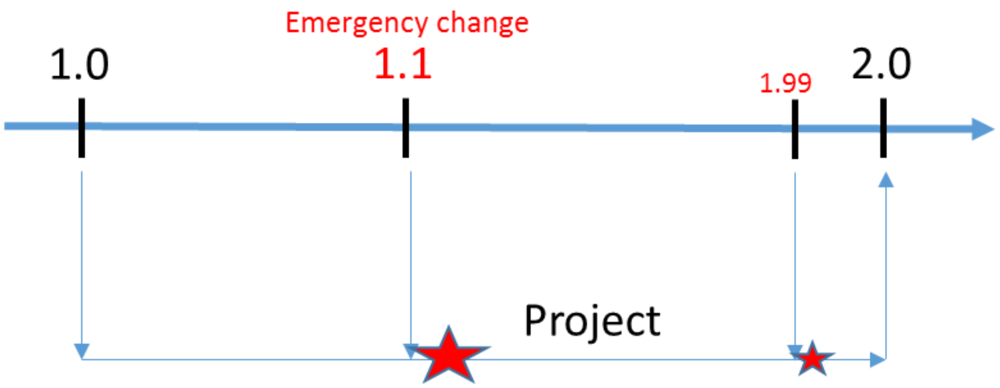


Initially, the project is based on baseline 1.0

When an emergency change (1.1) occur, the project must *rebase* to baseline 1.1. After the rebase, the project is based on baseline 1.1, both conceptually and actually.

Conflicts between project changes and emergency changes will be resolved by the project and on the project track.

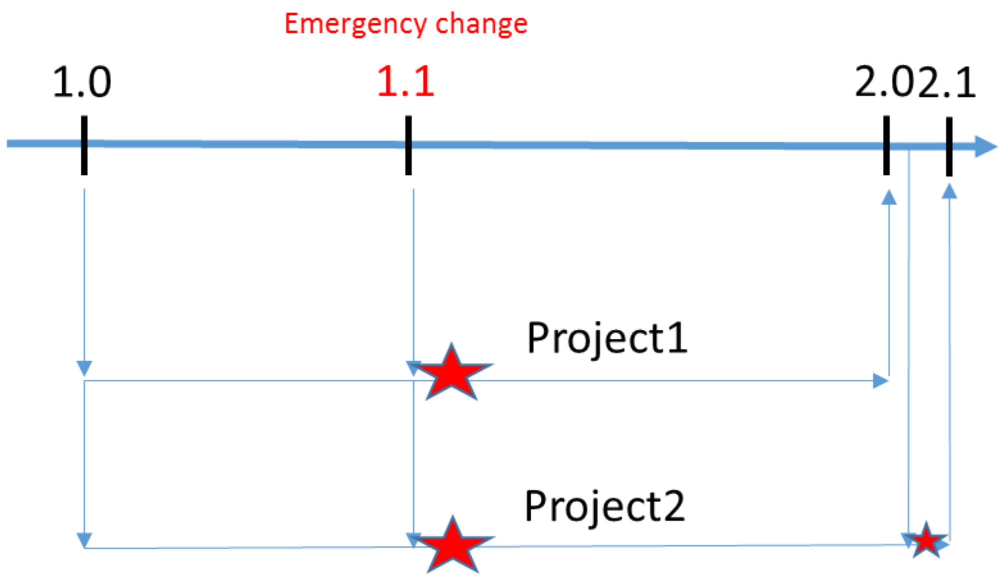
In fact, a project is forbidden to merge with the main track unless it is based on the latest baseline on that track.



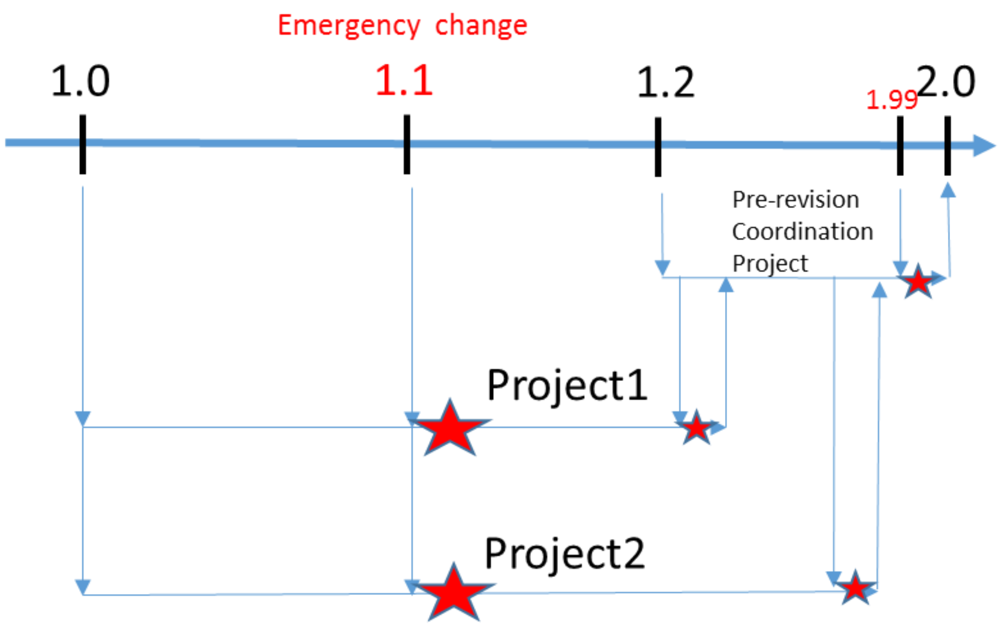
The main track is not necessarily a static configuration, especially not when dealing with a live installation (such as an industrial plant). Numerous changes can and will be introduced as part of day to day operations. Therefore we belive it to be important to do a ”safety” rebase just before merging the project to the main track. This may possibly require establishing a ”pre-merge” baseline to uniquely identify the pre-merge state.

A project is forbidden to merge with the main track unless it is based on the latest baseline on that track.

Continuation: Example rules for establishing and updating baselines in concurrent projects



Several concurrent projects can be managed by following the same principles. One remaining challenge will be to coordinate projects so that there will not be a race condition at implementation time.



One idea is to introduce a Coordination project to collect and coordinate the various concurrent project deliverables before revision time. This has the benefit of allowing more time for individual projects to deliver and to resolve inter-project conflicts, since the coordination project may be started well before the actual revision.
Ideally, the Coordination project shall have access to a validation and verification environment, if at all possible.

CCB practices

The concerned stakeholder interests must have appropriate representation at the CCB (or similar). Participators must have adequate competences so all relevant aspects of a suggested changed are put forward before decision. Responsibility to screen change requests and appoint appropriate reviewers should be assigned to a role (e.g. the CCB chairman or the configuration manager). It is also important that the participators take responsibility to look outside their own scope to support and understand the comprehensive view of the suggested change.

Scenario: CM in a power plant site

The following is an example of how the earlier described ideas about configuration ownership and change control can be put into practice, looking at the situation at a power plant site (see figure below):

The CEO has the overall responsibility for the site. The responsibility is delegated to a shared responsibility between the Directors of technology, Project, Planning and Revision, Security, Operation and Maintenance (in the figure on the next page these roles are referred to as NT, NR, NS, ND, NU).

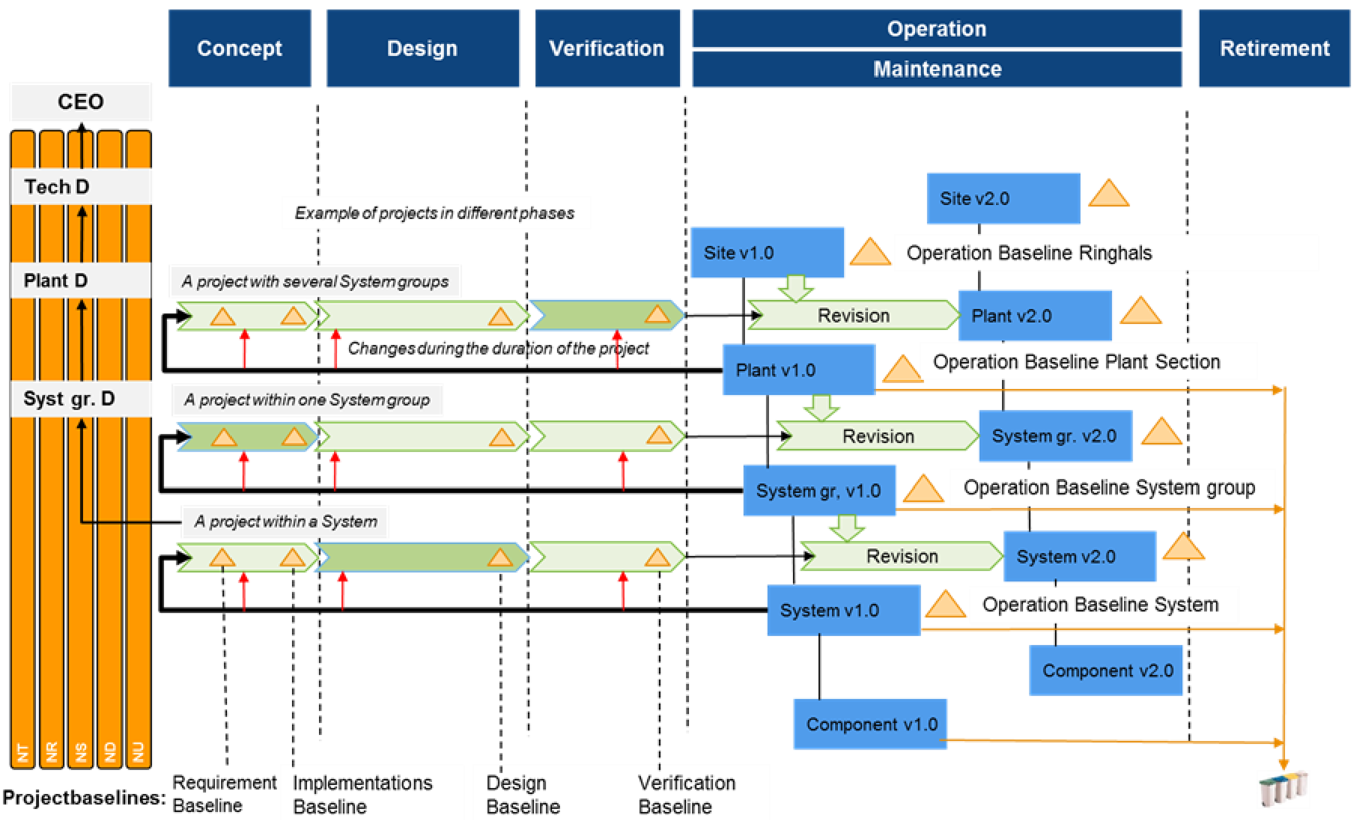
The Director of technology holds the combined configuration ownership, including the responsibility that the aspects from Project, Planning and Revision, Security, Operation and Maintenance are documented in a way that reflects the site's actual and planned states and are in accordance with regulations and requirements.

As the site contains four plants, the configuration ownership is delegated to the four Plant directors, who are responsible for the configuration of each plant with support from the Directors of technology, Project, Planning and Revision, Security, Operation and Maintenance. Within each plant, the configuration ownership is delegated to the system group Directors who are responsible for the configuration of the system group with support from the Directors of technology, Project, Planning and Revision, Security, Operation and Maintenance.

Suggested approach for CM

Typically, the change of configuration is pursued by projects passing through the phases as in the figure below.

To improve the control and lower uncertainties when implementing changes of the configuration, a structure of baselines that reflects the configuration ownership could be introduced. This enables freezing the configuration at a given point in time on the levels Site / Plant / System Group / System / Component, which in turn facilitates status accounting of the ongoing and planned changes to the configuration against particular baseline versions. Thus establish-



ing a comprehensive view of the results of the overlapping projects.

To secure a controlled change of the site's configuration over time, it is also necessary to work with planned future configurations that describes future states at defined dates (in one year, three years, five years, etc.). As the project's deliveries are the planned against these future configurations, a view of the effect of all ongoing projects can be established. When a new project is planned, the current affected baselines are the assigned starting point for the project and the defined project result is included in the applicable future baselines. The dependencies to other projects and ongoing changes are this way controlled by means of planning, defining and updating baselines.

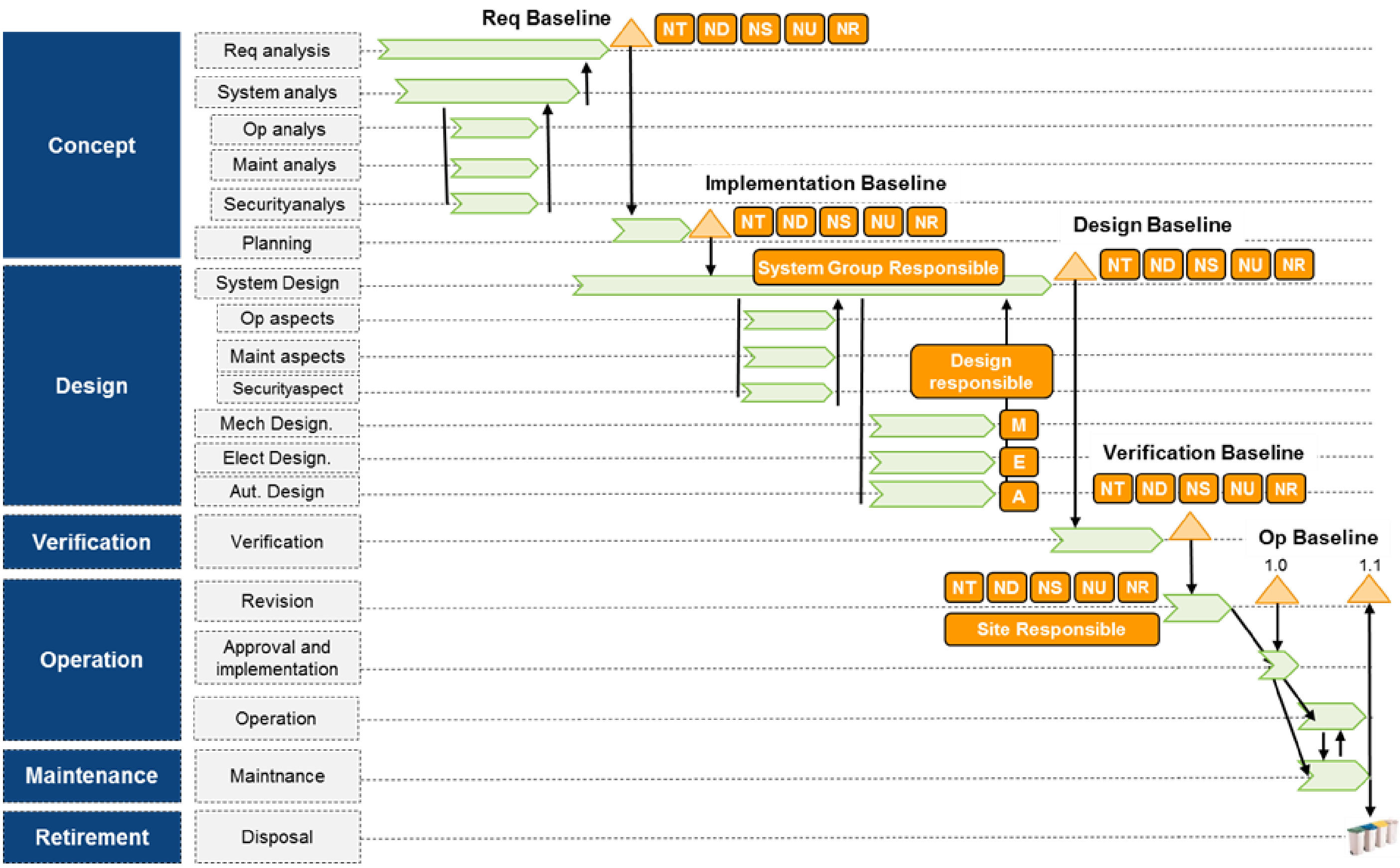
As long as change management is applied against both current and future versions of defined baselines, the comprehensive view on how projects deliveries affect the configuration can be upheld.

To improve control of the project dependencies and the accumulated effect on the site configuration project baselines can also be defined. The ownership of these would be analogue to the configuration ownership described above, but extended with responsibilities defined for Design, Mechanics, Electricity and Automation for the design phase and plant reasonability for the verification phase. See the figure below.

The starting point is in the concept phase has two types of baselines:

- The Requirement baseline that contains the requirements and regulations set on the project. This includes requirements from security, operation, maintenance, affected system groups and other relevant stakeholders.
- The Implementation baseline which contains the project deliverables. The Implementation baseline also constitutes the basis for the design work.

When the project is finished with the design of the new operational physical configuration a design baseline is established. The changes against this baseline are verified and planned for implementation by the revision process. After all physical changes, software's, procedures, training and documents is updated and approved according to the decided new configuration the change is fulfilled.



CM in an organizational context: *Positioning CM responsibilities*

How do you position CM responsibilities in the organization to establish clear mandates? How do you establish an active ownership of CM that allows for flexibility and participatory decision-making? How do you define the role of the configuration manager compared with the individual responsibility to apply CM practices that lie on each co-worker?

When establishing The CM organization, one should consider whether to establish a CM line or to distribute CM resources over the different departments.

If a CM line is established, it should be placed within the System Engineering or the Project office, or within the Technology department, in parallel with other units, but independent. There should be a CM Line Manager appointed. Advantages with a CM Line are:

- Easy to exchange experiences and knowledge.
- More clear definition of the CM role.
- Easier to get support if some CM related complications show up (i.e. CM role issues, work load etc.).
- Better possibility to make changes and improvements.

If instead resources are to be distributed across departments, awareness and knowledge must be kept alive by means of an internal CM network. Advantages are:

- Better prerequisite for understanding the respective product.
- More close relationship to project members (as they belong to the same organization).

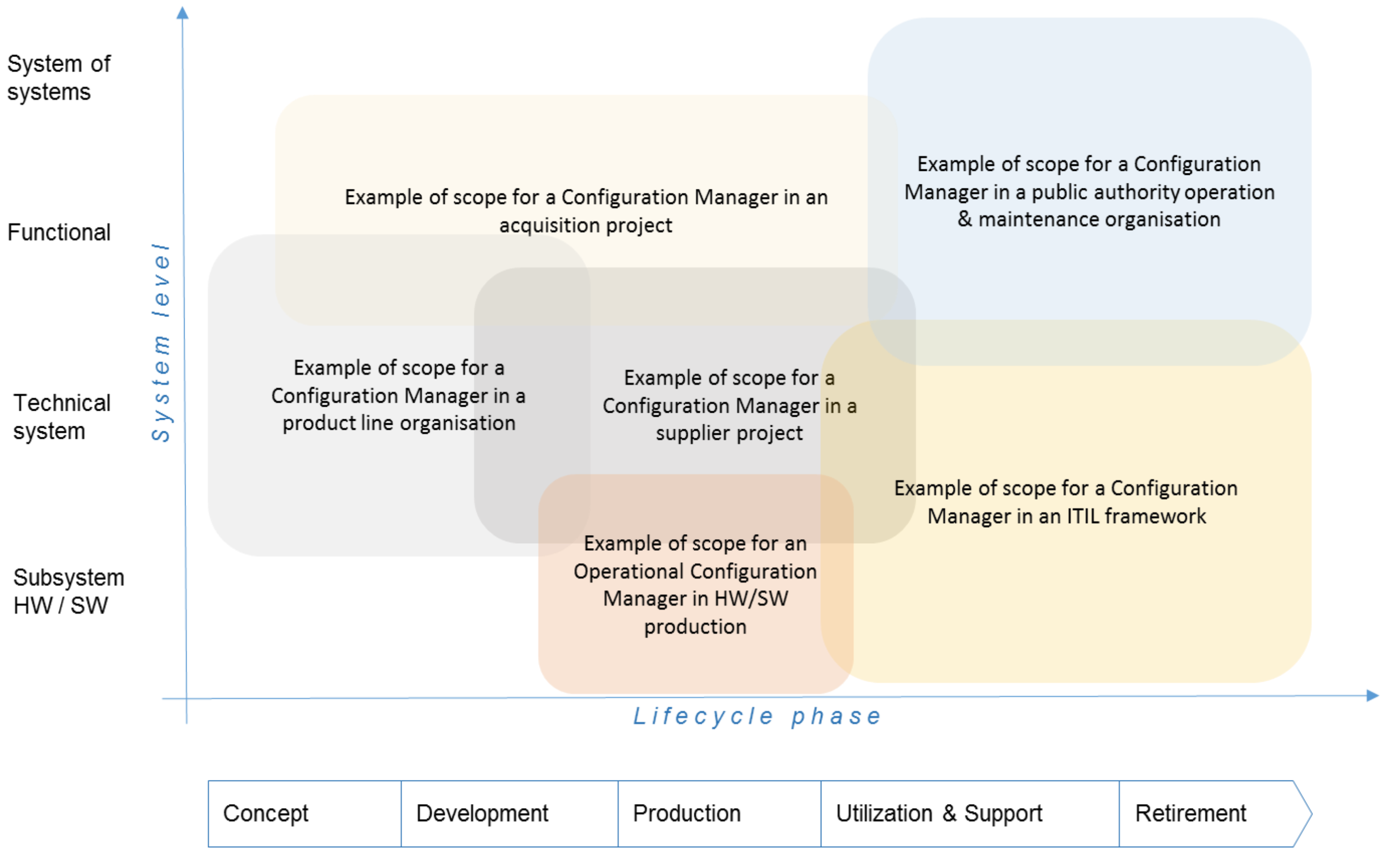
In some organizations, overall responsible for CM strategies are assigned to the role Strategic CM, this role can be combined with a CM Line Manager. The strategic CM owns the CM process and works long term to formulate and pursue change. The Strategic CM is available to support and coach the CM organization.

The role of the Configuration Manager

The scope of the Configuration Manager role is first and foremost dependent on the terms of reference for CM. In organizations that do not have a general CM regulation or policy, the terms of reference for CM is often defined within a project scope or the frame of a contract. For these cases, the role of the Configuration Manager is defined from what is needed for the particular project or contract. For large organizations that develop and manage complex systems (products and services), the scope of the Configuration Manager is usually limited to a defined system of interest and a particular part of the product lifecycle. This is due to how divisions, departments and projects usually are scoped.

The terms of reference for CM is to make sure that the products configuration and product configuration information is:

- Planned with baselines against the product architecture and lifecycle
- Identified
- Reviewed and audited



- Formally change controlled
- Status accounted

To achieve this, the CM function must set up:

- Processes and methods
- Rules and instructions
- Role descriptions
- IT to support the above

The Configuration Manager should:

- Work closely together with project management to understand the project cycle, milestones and objectives.
- Work closely together with the system architect to understand the effects of identifying Cis and the version/ variant principles.
- Work closely together with the product owner and the development organization to understand the product, its lifecycle and release planning.
- Cooperate with, and support designers.
- Cooperate with, and support aftermarket & maintenance organization.
- be visible and available.
- be part of new projects from the start, i.e. already from customer negotiations.

The figure on previous page exemplifies the scope for operative Configuration Managers. Along the figure Y-axis, the system hierarchy increases, starting with hardware or software components and ending with high-level system of systems. Along the X-axis is the system lifecycle.

The opposite of attributing mandate and responsibility regarding CM to a particular role, the Configuration Manager, is to let this mandate and responsibility be totally distributed to all the co-workers and other adequate roles, such as Project Manager, Administrator, Document manager etc. With totally distributed configuration management, rules and routines for CM are clear, known and followed by everyone. IT-tools are configured and set up in a way that supports CM activities.

Even if the role Configuration Manager exists in the organization, all co-workers still have a CM responsibility. The Configuration Manager should first and foremost be responsible of making sure that the applicable regulations and routines are followed.

Typical responsibilities for an Operative Configuration Manager:

- Writing CM Plans or Document Plans.
- CI identification and naming.
- Start-up working areas for the project.
- Manage structures.
- Manage baselines.
- CCB and Change control.
- Administration of CM tools.
- Handling releases (including writing PRI).
- Handling deliveries, both internal and external (create, deliver and follow up) .
- CM status reporting.
- Perform audits.
- Support/educate team-members in CM issues.
- Review document formalities.
- Follow up on CM routines and regulations.

Typical responsibilities for a Strategic Configuration Manager:

- Process responsibility (improvements and implementation) of change control, release management, document management, etc.
- Maintain the generic CM plan.
- Raise CM awareness within the organization and share CM knowledge.
- Enforce that CM processes and methods are followed.
- Improve CM processes.
- Plan and decide on CM strategies.
- Set requirements on IT-tools for CM.
- Networking internally/externally to develop CM.
- Ensure overall conformance to legal aspects.