# Scope of this Document

Blue Dolphin has a long-standing history of successful Critical Chain Project Management (CCPM) implementations.

Nevertheless, it’s always a hurdle to convince an organization to start. Main hurdle is that the effects are not qualified before the middle of the change – because you need good CCPM-Like Project plans to identify the constraint and do a correct staggering.

“Essential FLOW/CCPM” focuses on the identification of the constraint just by using rough planning data and doing the staggering based on a Simplified Drum Buffer Rope-Style just at the constraint. This also enables a very rough phase gate-based execution management and a rudimentary buffer regain / process improvement process. This all without having a detailed project planning.

If it is necessary to manage bigger projects with external dependencies its no big thing just to do this in a CCPM-style in parallel according to the staggering of the Essential Flow/CCPM. So, there is a natural lightweight migration path to full blown CCPM.

The only drawback is that Essential Flow is more focused on the optimal usage of the constraint – so the focus is throughput. To keep the system stable and the simplified DBR valid there has to be more buffer than in full blown CCPM – so the final optimization of the lead time stays the domain of the full blown CCPM.

This document describes the Concepts behind “Essential FLOW/CCPM”.

# Credentials

The core idea was developed by Wolfram Müller and Steve Tendon in a project to speed up delivery in a 4500 people developer company. Based on Tame the Flow and Simplify Drum Buffer-Rope (sDBR) Concept.

This work was based on the Papers of Eli Schragenheim and Rudi Burkhard about the Simplified Drum Buffer Rope including Rapid Response Capacity. And this furthermore was based on the groundbreaking work of Eliyahu Goldratt and the Theory of Constraints.

The reference Implementation of the sDBR and the underlying paper can be found in the information package available s. <https://reliable-scrum.com>

Additional thanks also to Jan van Egmond and his “Empower Your Project” who made us think about reactivating these ideas to lower the entry barrier for modern Critical Chain Project Management (CCPM) implementations s. <https://empoweryourprojects.com/>. Furthermore, he had the idea to enhance the 1/3 red/yellow/green execution management from the sDBR to a little better phase-based lead/in constraint/lag execution management. In practice the difference is minimal, but it helps to focus the organization to point the attention always on the constraint and the optimal usage.

# Essential FLOW/CCPM

Essential FLOW/CCPM covers three important phases of managing a multi-project environment.

1. Module A - Identifying the Constraint  
   initially determine the real or virtual skill that constraints the overall output of the system
2. Module B – Staggering of the Projects  
   determine the start- and due-dates of all projects, the ones with constraint usage and all the others – these are the real dates to be committed
3. Module C – Execution Management and basic Buffer Regain Management  
   monitoring the status and phase a project is in and should be, keep an eye on the measures to get the projects back on track

## Module A “Identify the Constraint”

s. Excel-File “Module A - Constraint Detection“ is just once in the beginning or if is looks like that the constraint has moved.

### Input:

* A list of real or virtual skill with their capacity. The capacity is typically “net available working days per month” – we use typically 205 working days/year and a project quotation of 60-80%.
* For each of the skill groups there is a rough estimate about how many people at maximum can work on one project at the same time. This is used not for identifying the constraint but for estimating the duration of a project in the constraint for the later staggering (s. Module B).
* A matrix of all projects and their planned load (in same unit as the capacity). It does not matter whether the projects are started or not and the accuracy has to be very rough – our experience is that a constraint is so strong that it really stands out of the mass. In case that there are two or more equal constraint candidates these load data have to be checked.
* The list of the projects is ordered – means top priority first. Very often the priority is shown as a priority level.

### Output:

* The loads over all projects are summarized for each skill. Divided by the capacity of the skill you get the “queue length”. The skill with the longest queue is probably the constraint.
* Typically, the skills are sorted from left to right by their length of the queue to easily get the overview.

### Typically, Situation (real Example)

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

On the left in green the project list ranked top priority first

In the columns D and following in the header area you see the skill groups, their load = aggregation of all loads in the matrix below, the capacity per month and the resulting queue length.

In the matrix starting with D9 there are all the loads per project and skill group.

### Enhanced Output

Out of the data s. above you can easily derive the project list for the staggering. For this list you need the duration in the constraint. This can be wildly guessed by using the load information and some assumptions on how many people can work on a project in maximum.

In practice there are always projects that are not bound to the identified constraint. In this case the overall duration of the project can be roughly estimated by looking at the load of the most needed skill group for this project.

The following table shows how such a heuristic to estimate the durations can look like.

For sure after the estimation the responsible people have to check this and adjust it to the current reality e.g. some project are partly done.

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Project list with information whether the project is actively planned and does use the constraint or not.

If it uses the constraint, it shows the constraint team and the capacity, and the maximum capacity used in a project – in FTE (Full Time Equivalents).

Based on what available FTE is smaller a rough duration is calculated – either for the constraint duration or for the overall duration.

This table can be copied and used for the Module B “Staggering”.

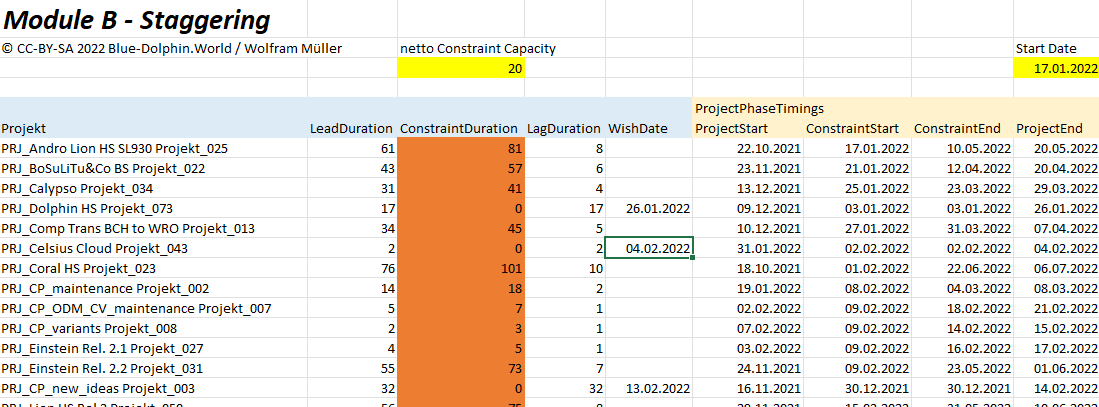
## Module B “Staggering”

### Input s. list above – output of module A.

* It’s mainly a ranked project list with the information: duration before the constraint (lead), duration in the constraint, and duration after the constraint (lag).
* Additionally for projects not determined by the constraint a wish-date that is used as due-date.
* A rough estimation of how many projects can be simultaneously in the constraint without negative multitasking.
* A start-date for the planning / timeline / Gantt-Diagram.

### Output

* For each project the Start and Due-Date. For projects not using the constraint the wish-date is used as due-date.
* For each project using the constraint the date when the project should enter and leave the constraint.
* Typically, there is a visualization as a Gantt-chart with the time in the constraint shown in a alarm color.



The Project list blue with the date yellow.

The dates are calculated based on the queue length in the constraint of all projects before the current project with regards to the working days:

* ConstraintStart = StartDate + WorkingDays( Sum of all ConstraintDurations before / netto ConstraintCapacity)
* In case the project doesn’t use the constraint the imaginary ConstraintStart = WishDate – WorkingDays( LagDuration )

Based on this all the other dates are easily determined:

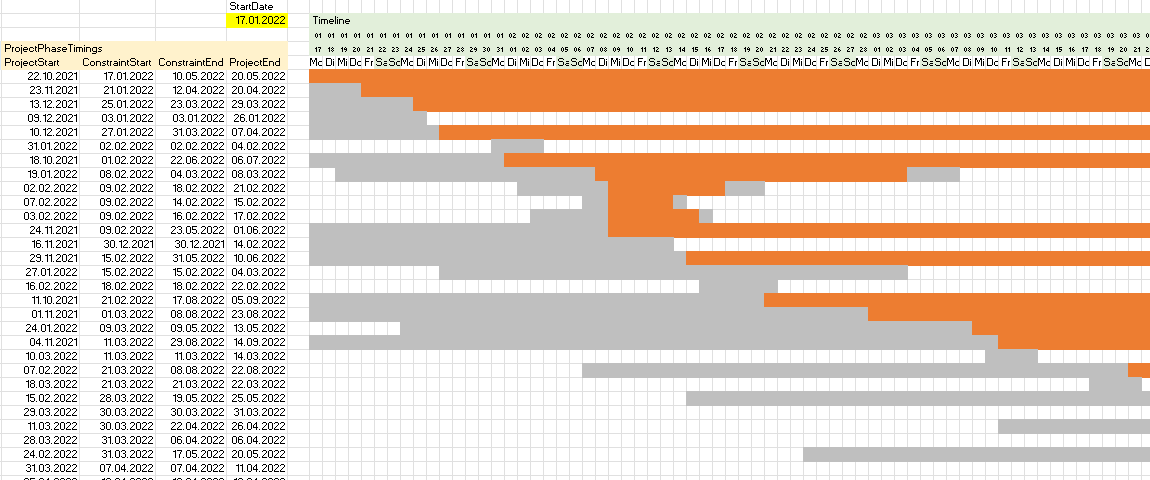
* ConstraintEnd = ConstraintStart + WorkingDays( ConstraintDuration )
* ProjectStart = ConstraintStart – WorkingDays( LeadDuration )
* ProjectEnd then = ConstraintEnd + WorkingDays( LagDuration )

With WorkingDays() as the build in Excel-function moving a date forward or backward in time by recognizing weekends.

Here the usability is very important – it must be easy pull project up- and down in the list. The Dates will change and the Gantt chart to. It may be interesting to do this in a simulation mode so that the “released” planning is not changed.

Projects using the constraint are scheduled solely by the constraint usage. Therefore, it can be that the ProjectStart can be earlier as the timeline StartDate. This fits typically perfect to the reality, because all high-priority projects should be on top of the list and therefore already critical according to the due date and therefore they should be in the constraint. In case they are not in the constraint yet – you can easily move the project down in the list until the real lead duration is shown.

The resulting roadmap is often shown as Gantt-Diagram with the time in the constraint highlighted:



## Module C “Execution Management”

Based on the phase start and end dates combined with the current phase status it’s easy to determine roughly whether a project is on track or not:

### Input:

* The ranked project list with all phase start- and end-dates out of module B “staggering”
* The current status in which phase the project is currently in. We distinguish:
  + NS for “not started”
  + BC for “before the constraint”
  + IC for “in the constraint” if applicable
  + AC for “after the constraint”
  + FN for “finished”
* The date of the status to determine in what phase the project should be.

### Output:

* In case the current phase is lower than the planned phase then there is a message/hint to speed up
* In case the project is started before the ProjectStart there is a “stop now” message

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

### Open Issue List

For each project with a message there should be a measure how to bring this project back on track – a classic open issue list:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Typically, one collects the measures and root-causes in form of a logbook as input for a kaizen session to improve the process in the long run.

# Further Ideas to Essential FLOW/CCPM

* The lead/in constraint/lag-structure with the appropriate determined dates could be the frame for a detailed projects plan – even with CCPM buffer management and fever curve.
* The current staggering assumes full availability of the constraint for regular projects. The staggering formular can easily be adjusted to use a part of the capacity for “Rapid Respons (RR)” projects as shown in the reference implementation of the sDBR.
* The execution management can be enhanced by asking about the remaining duration in the current phase. Then some kind of a rough forecast can be used to determine the probability of meeting the next phase start date (e.g. ConstraintStart).
* The results of the staggering can be tagged as scenarios – out of these tagged scenarios a decision paper can be generated s. 4-6-4-Methode of Wolfram Müller
* Normally there is no need for having a project plan. In case the dependencies or critically or complexity makes it necessary – it would be cool to have a button that translates the data out of the Essential CCPM world into a full blown CCPM plan. The phases and dates and even the load of the constraint could be used to set up an initial CCPM plan with three phases and a work package in each phase connected. The project manager can work in the CCPM plan and the debuffered data is transferred to the Essential CCPM forth and back.