

# Why engineering simulation models configurability and flexibility matter

Webinar on 2022-10-26 by:

**Patrick Cloutier, P.Eng**   &   **Vincent Béchard, M.A.Sc.**

[patrick@simalking.com](mailto:patrick@simalking.com)

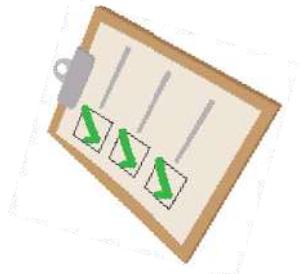
[vbechard@difference-qcs.com](mailto:vbechard@difference-qcs.com)



# Agenda

---

- ≡ What is operations simulation?
- ≡ Why is it important to engineering?
- ≡ What makes it even better: configurability!
- ≡ A flexible and limitless platform
- ≡ Example 1: scheduling for a food processing plant
- ≡ Example 2: material handling and traffic debottlenecking
- ≡ Final words



# What is operations simulation?



# Why using a model?



Asset condition monitoring and increase of uptime

Methodic de-bottlenecking and economic ramp-up

Locate, track and quantify wait times & unproductive delays

Develop smart solutions for mines, plants, buildings, cities, ...

Forecast demand and anticipate productivity

Improve scheduling & supply chain efficiency

Diagnose and optimize process controls

Democratic information sharing between people & systems

Real throughput?  
Best schedule?  
Enough resources?  
Deliver on time?  
Reduce costs?  
Scale up?

# Typical model elements

---

- ≡ Operations simulation models capture all aspects of a production system to help analyzing and improving the global operational performance:
  - ♦ **Schedules** daily shifts, weekly production, planned maintenance
  - ♦ **Equipment** capacities, reliability, buffers, storage, conveying, piping
  - ♦ **Operators** activities duration, task coordination, procedures
  - ♦ **Maintenance** spare parts, work orders, repair times
  - ♦ **Transport** mode, traffic, speed, loading times, queuing at intersect.

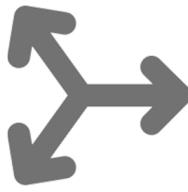


# About event-based modelling

Model systems which change states at discrete points in time as a result of specific events

## Examples of EVENTS

- Order/part arrivals
- Product movement
- Machine process start/finish
- Machine breakdown/repair



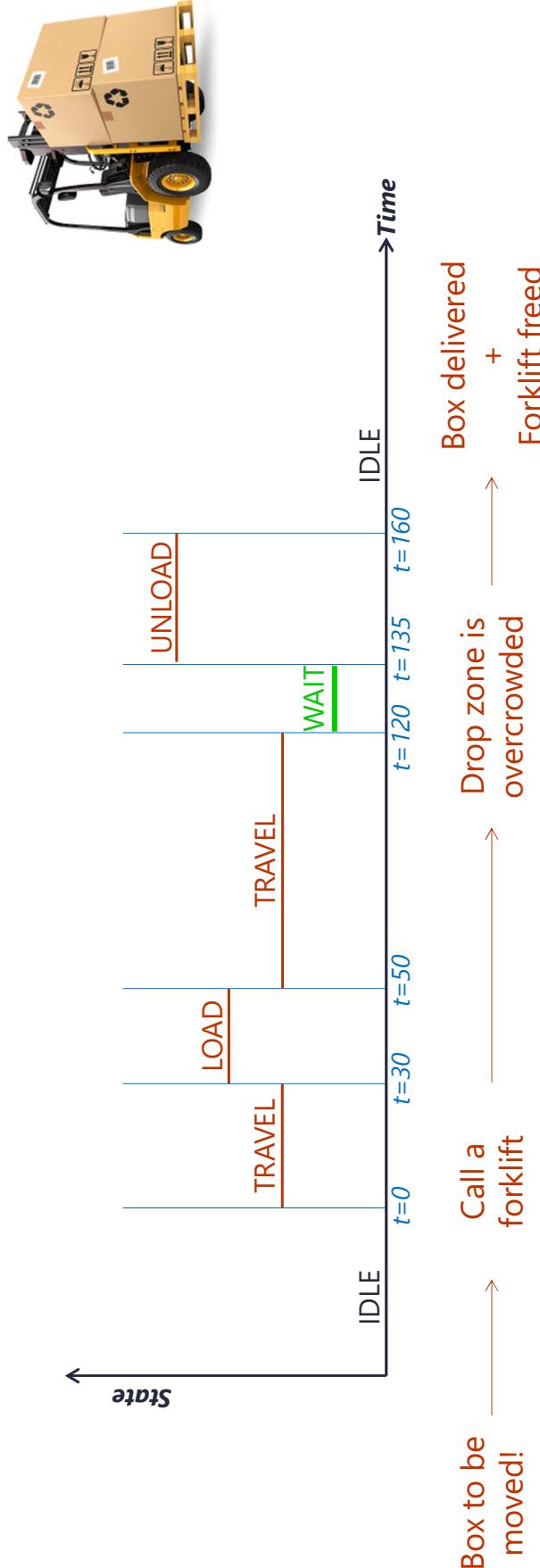
## Examples of STATES

- Machines: idle, setup, processing, down
- Queues: empty, full
- Operators: on-shift, off-shift, utilized, idle
- Transports: travelling, loading, unloading

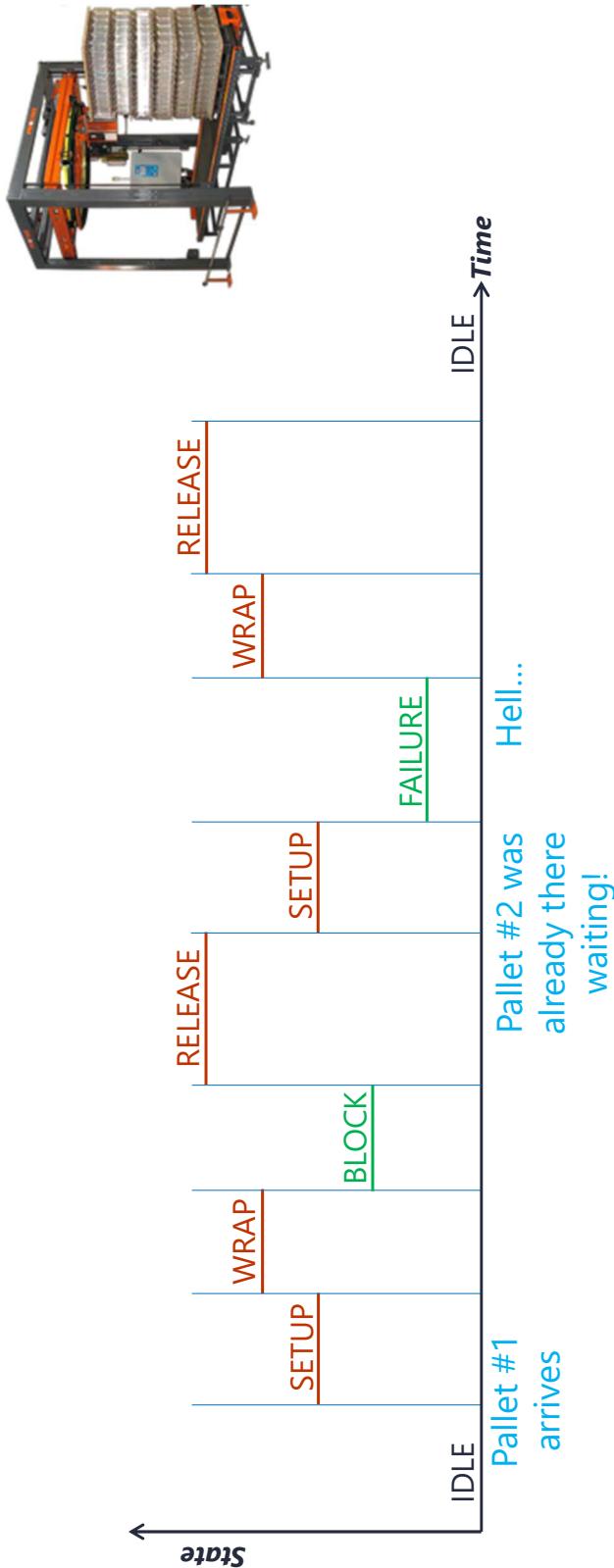
## Simulation QUEUE and CLOCK

- Event execution & state updating trigger future events
- All events are put in a time-sorted queue
- The simulator jumps to the next timed event and executes it

# Example: moving a box

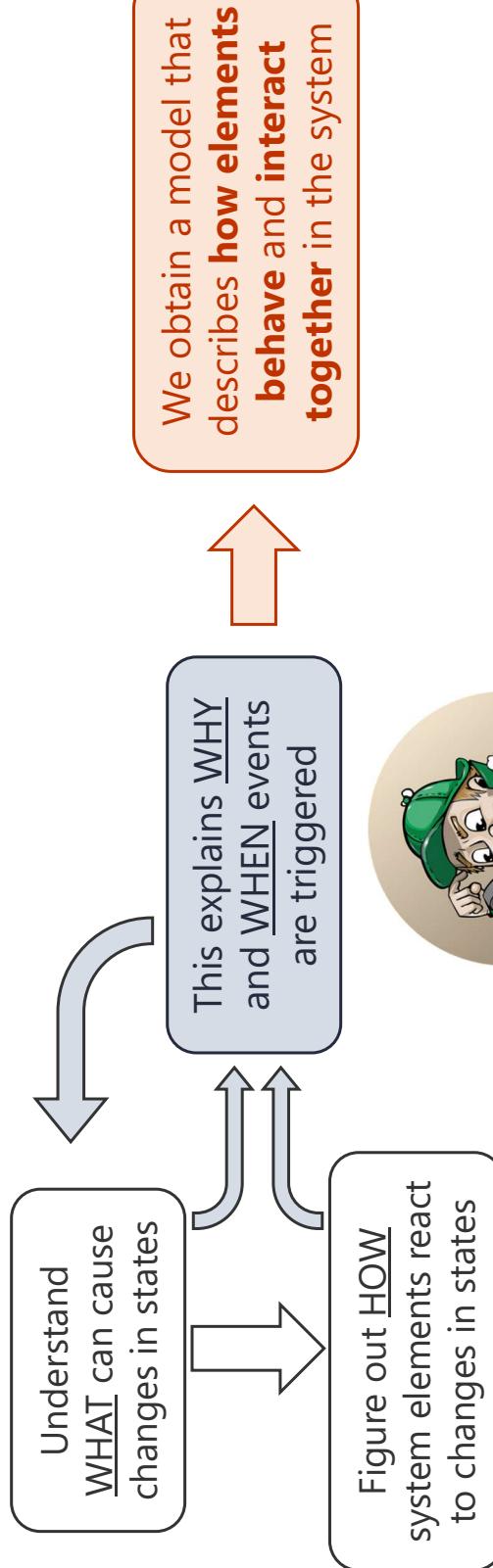


# Example: pallet stretch wrapper



# Event-based modelling

The mental exercise behind event-based modelling is:



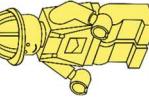
*Even without a computer implementation, investigating and developing this vision of a system is value added!*

# Building blocks of complex systems

*Most production systems and supply chains can be decomposed, modelled and optimized using these elements!*

## VARIABILITY

- ≡ Probability distributions
- ≡ Delays and rates
- ≡ Uncertainty tolerances



## MANAGEMENT

- ≡ Scheduler
- ≡ Dispatcher/regulator
- ≡ Assets tracking
- ≡ Performance monitoring



## DELAY

- ≡ Cycle time
- ≡ Time to failure
- ≡ Repair time



## WAIT

- ≡ Queue (FIFO)
- ≡ Stack (LIFO)
- ≡ Priority pick



## INVENTORY

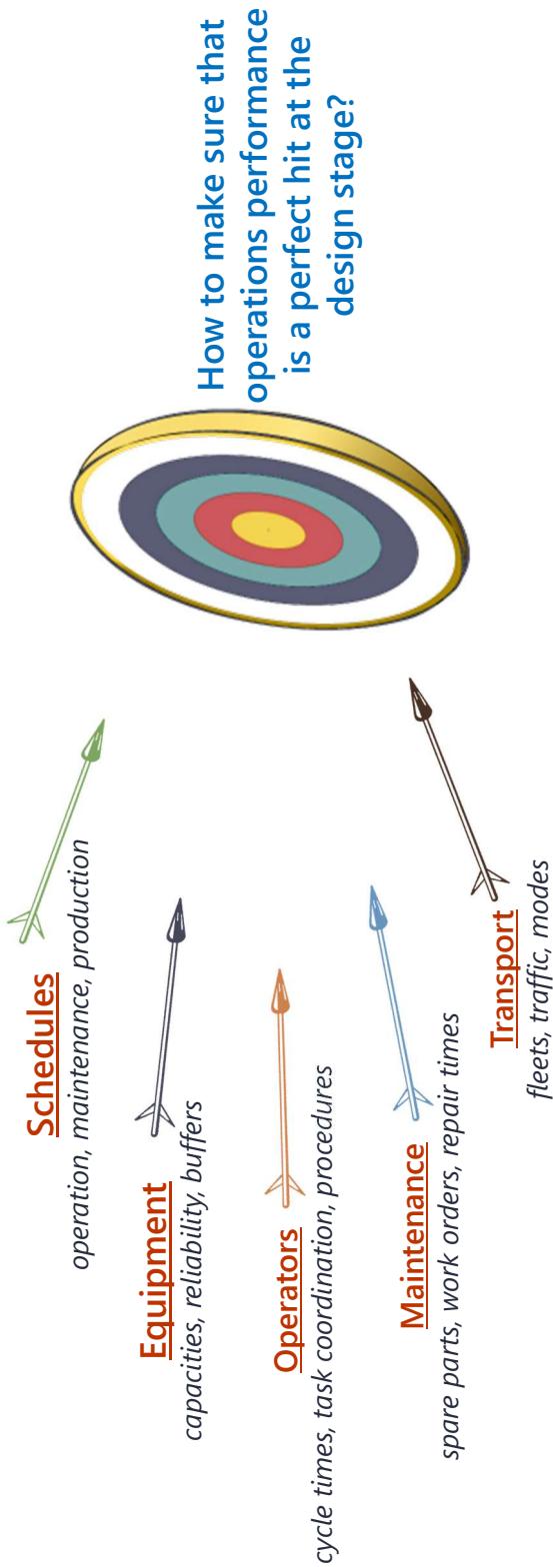
- ≡ Items storage
- ≡ Parts buffer
- ≡ Tank/silo

# Why is it important for engineering?



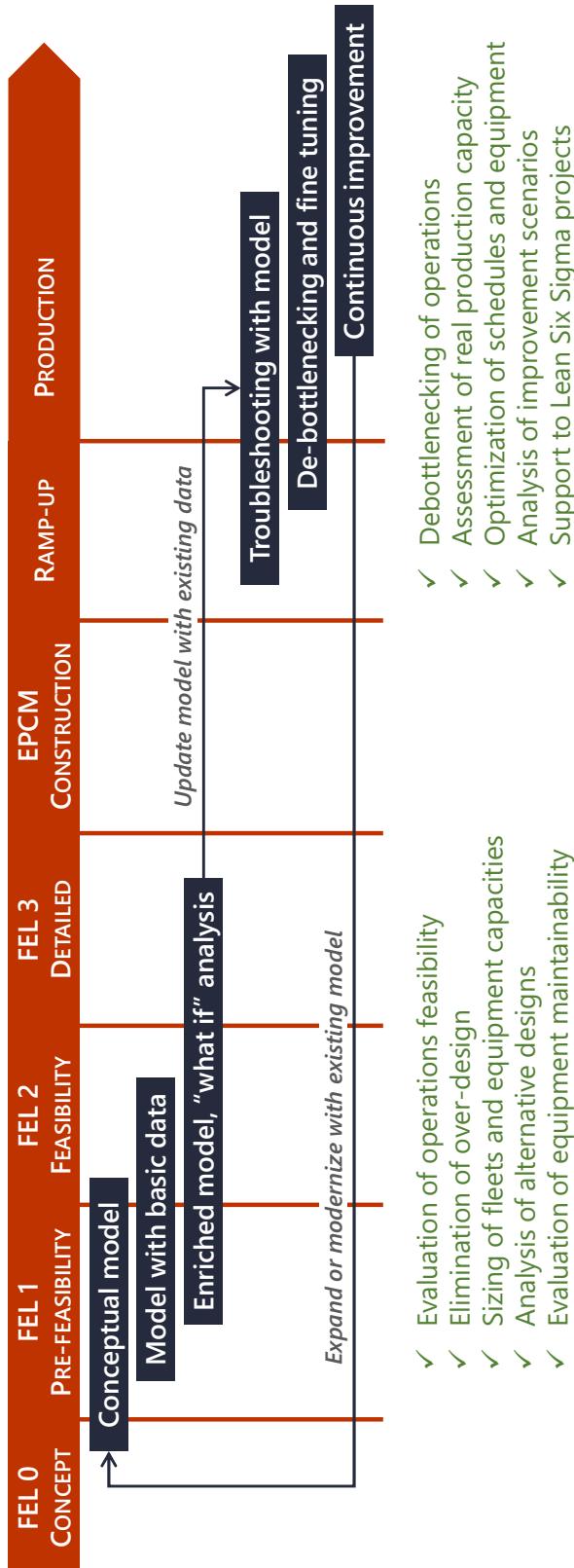
# Engineering design challenges

Obtaining the perfect system design that will operate smoothly and efficiently is not straightforward...

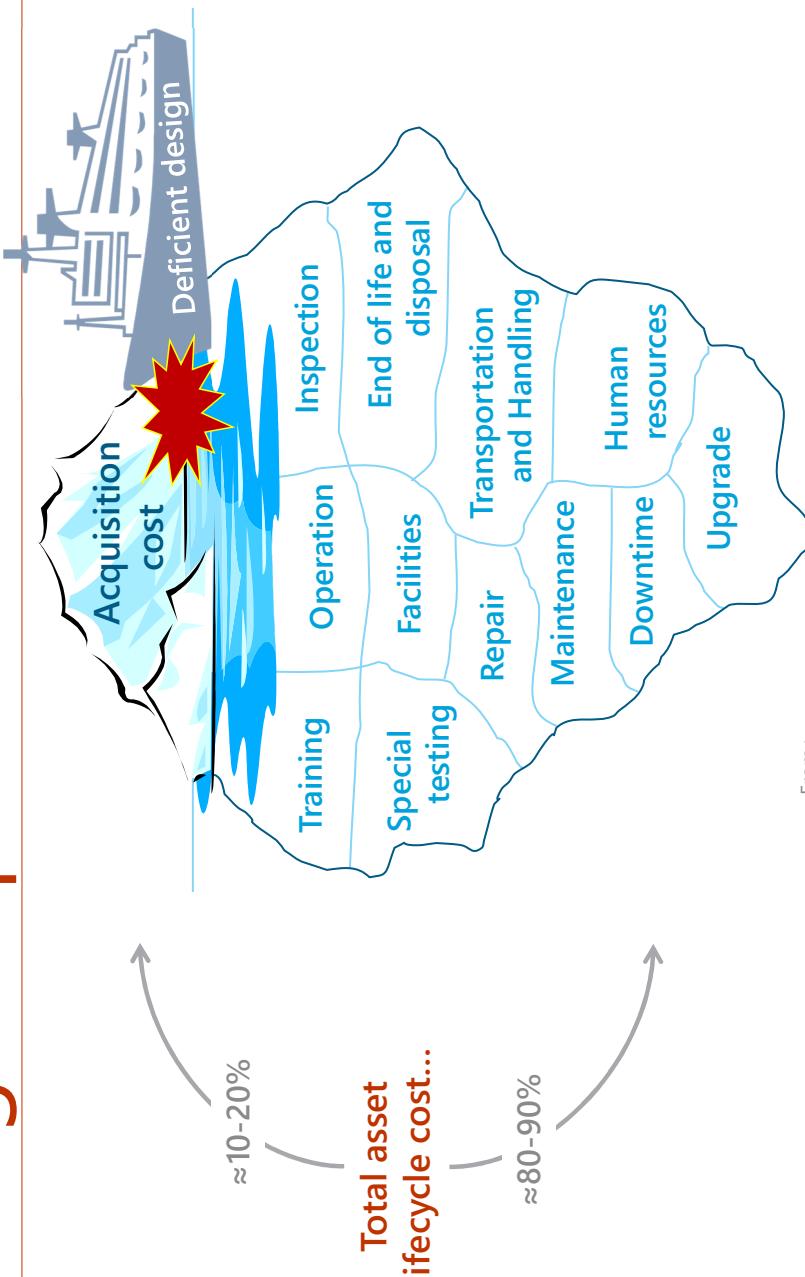


# Typical engineering design cycles

*Simulation should be used from early stages to increase the chances of a perfect target hit on operational performance!*



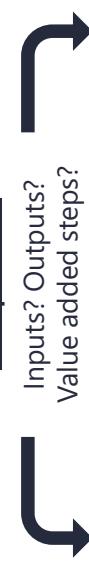
# Design impacts the total cost...



# Operational thinking during design

Operations simulation brings an operational thinking at the design stages to ensure the whole life usefulness

## The process



## Equipment

Quantities? Configuration?  
Capacities? Reliability?

## Layouts

Effective flows? Storage  
vs working areas?

## TYPICAL

ENGINEERING  
FOCUS



*Everything is interrelated!*  
*Are equipment and layouts optimized  
for operations and maintenance?*

# The perfect design testing sandbox!

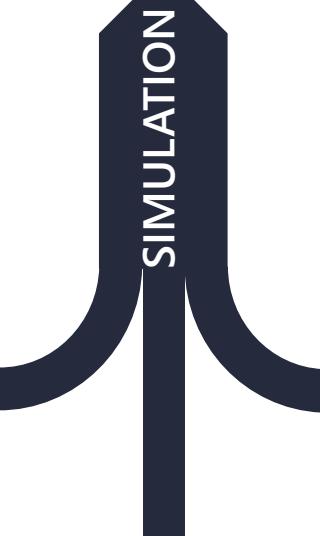
Layouts, routes, dimensions,  
stationary and mobile  
equipment characteristics,  
technical data

DESIGN  
SPECIFICATIONS  
*engineering*

HISTORICAL  
PROCESS DATA  
*data science*

Cycle times, schedules,  
resources availability, logic,  
routing and operational  
constraints, reliability

OPERATIONS  
KNOWLEDGE  
*people*

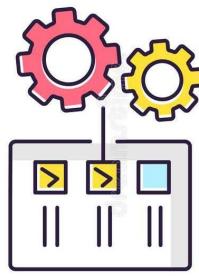


## Outputs:

- ≡ Performance diagnosis
- ≡ Effective and ultimate throughput
- ≡ Equipment utilization rates
- ≡ Meaningful and custom statistics
- ≡ Trend and pie charts, histograms



What makes it even better:  
configurability!



# Different modelling flavours

---

## Design sandbox

- ✓ Best estimator of expected assets
- ✓ System parameters TBD
- ✓ Resource counts TBD
- ✓ Layout and visuals TBD
- ✓ Purpose: determine the ideal system parameters (the design!)

## Digital twin

- ✓ Virtual duplicate of existing assets
- ✓ Fine details on system parameters
- ✓ Calibrated with recent data
- ✓ Hi-Fi interactive 3D visuals (immersive VR if possible)
- ✓ Purpose: understand the deviance from the designed base conditions

# Different modelling flavours

---

≡ Brings back to the fundamental question when starting a modelling task: what are the questions to be answered?

- Looking for a "designed base conditions keeper"?
- Need a Digital Twin!
- Looking for the system design itself?
- Looking for an engineering design sandbox!

# Designing a configurable sandbox

≡ Shifting the mindset is crucial!

Create a model,  
implement it and  
generate outputs



Deliver a scientific app  
implemented in a  
simulation platform

Traditional model development  
approach and usage by the  
model developer

Software-oriented mindset with  
planned user interaction (inputs) and  
feedback (decision-ready outputs)

# Model configurability in practice

≡ The secrets:

- Adopt an object-oriented modelling mindset
  - Break down the system in objects that will interact together
  - Objects population are easy to parameter: fleet sizes, number of processing lines, number of operators, etc.
  - Highlight which parameters are required by the objects to make decisions and to be configured
- Adopt a database mindset for the inputs
  - Before editing the model, think to all required parameters
  - Easy when an object-oriented model design exists...
  - Prepare a database schema of the inputs!

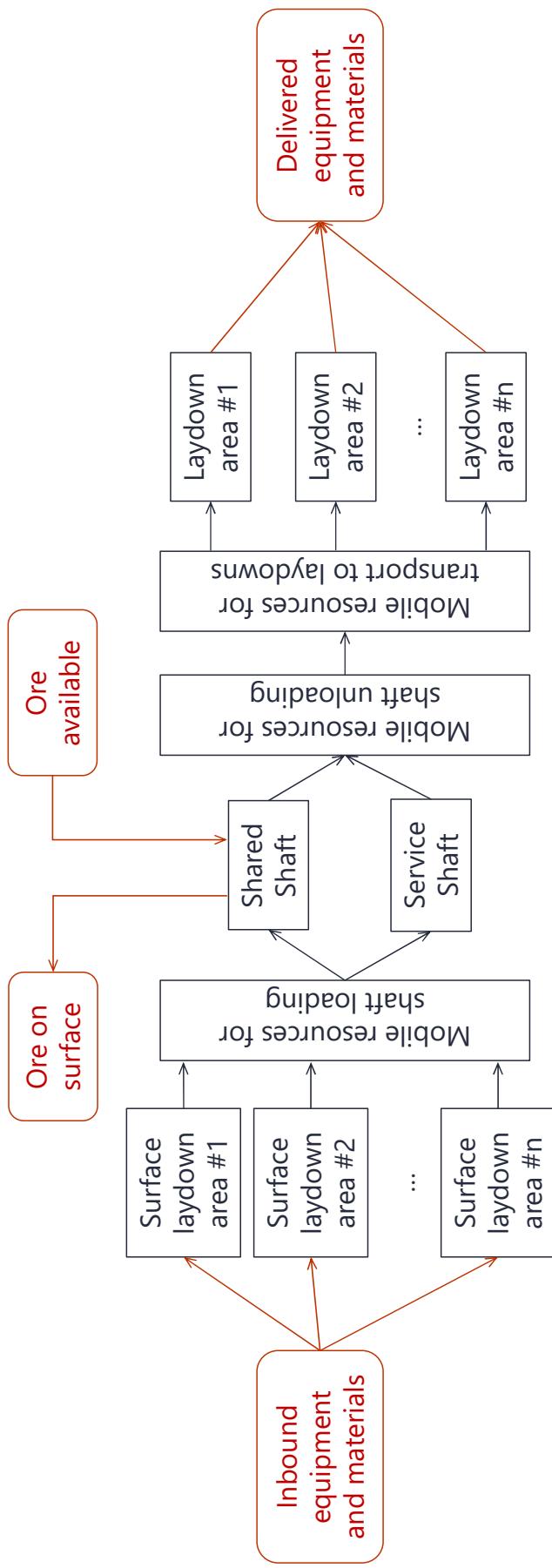


# Model configurability in practice

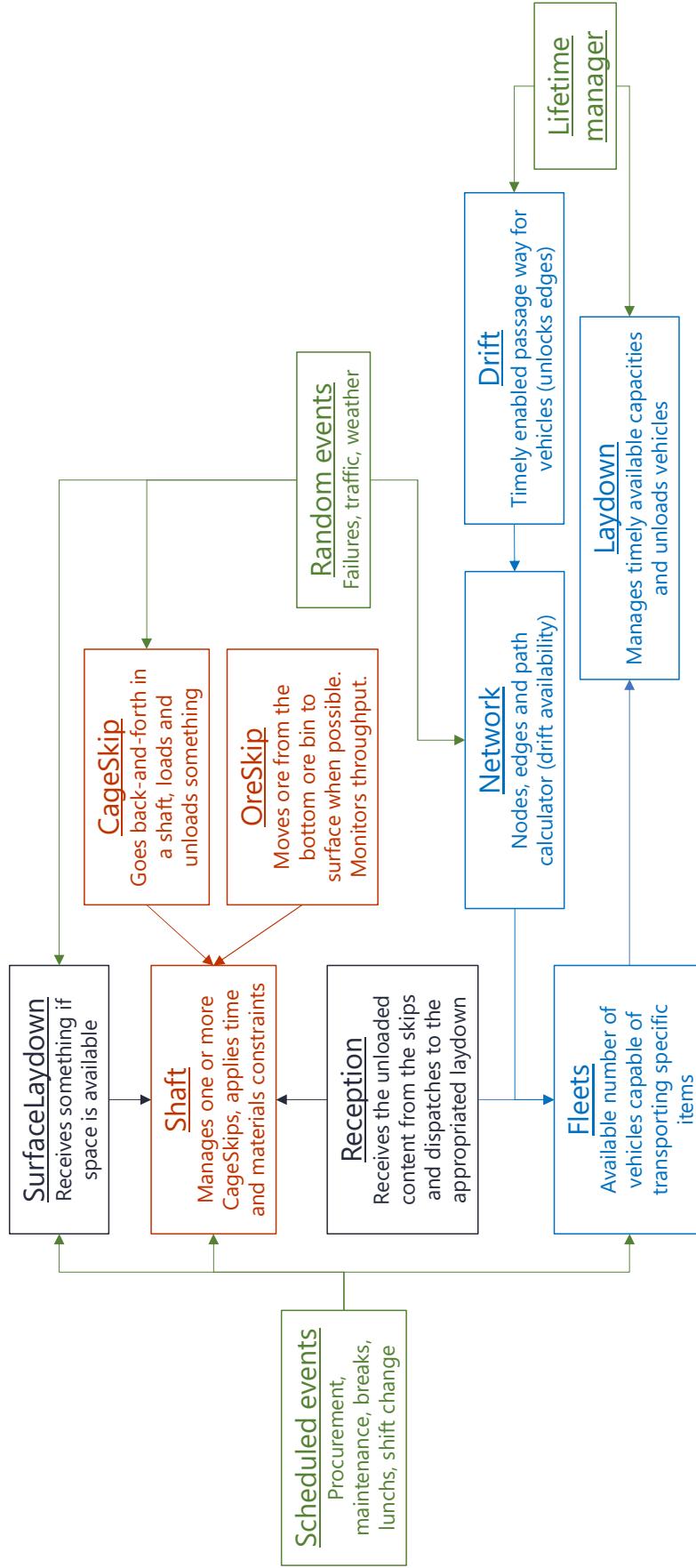
≡ Spend more efforts upfront:

- ◆ Before coding, prepare 3 diagrams:
  - The model scope: major activities, logical steps and streams, inputs to the model (the triggers), outputs (the end points)
  - The objects: the components of the model, their main roles-duties-features, their relationships
  - The input data: a database schema
- ◆ Store and manage the inputs with rigor
  - Garbage in, garbage out...
    - In Excel spreadsheets with data validation and dropdowns
    - In a database for stronger type checking and data indexing
- ◆ Prepare yourself to spend time managing the inputs
  - There will be scripts to write to parse the information
  - There will be scripts to write to construct the model on-the-fly

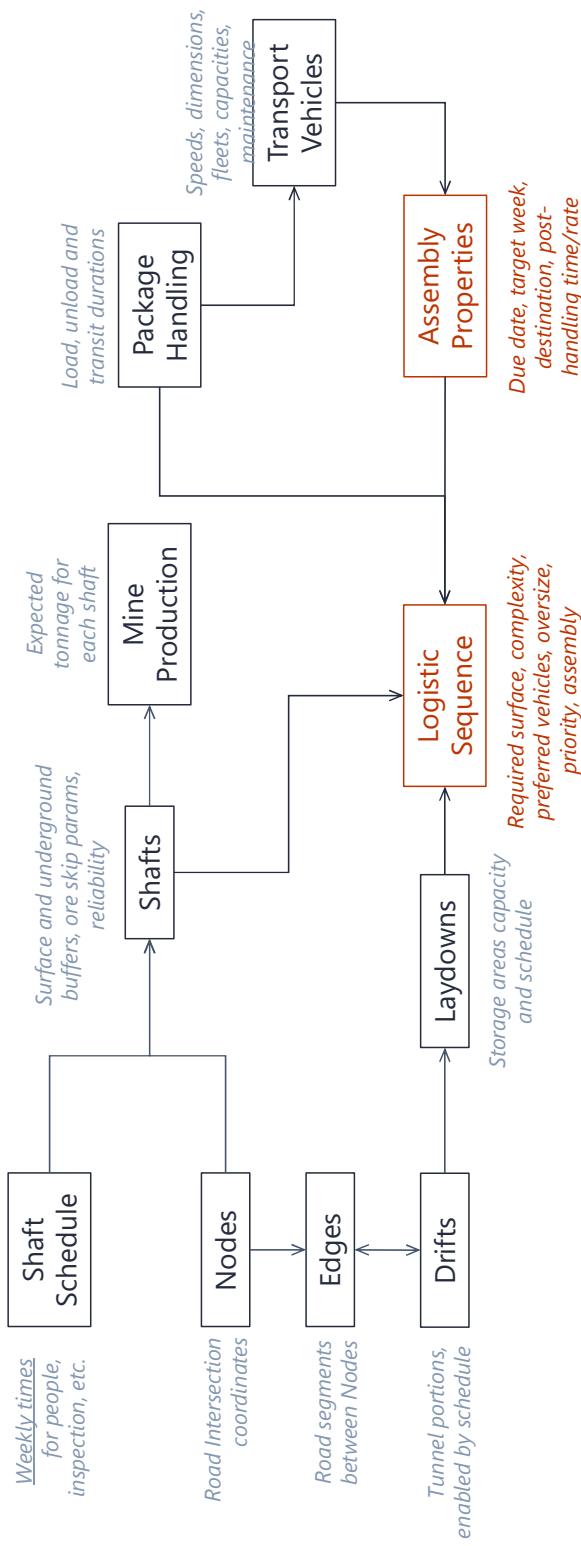
# Illustration: (1) model scope



# Illustration: (2) model components



# Illustration: (3) inputs diagram



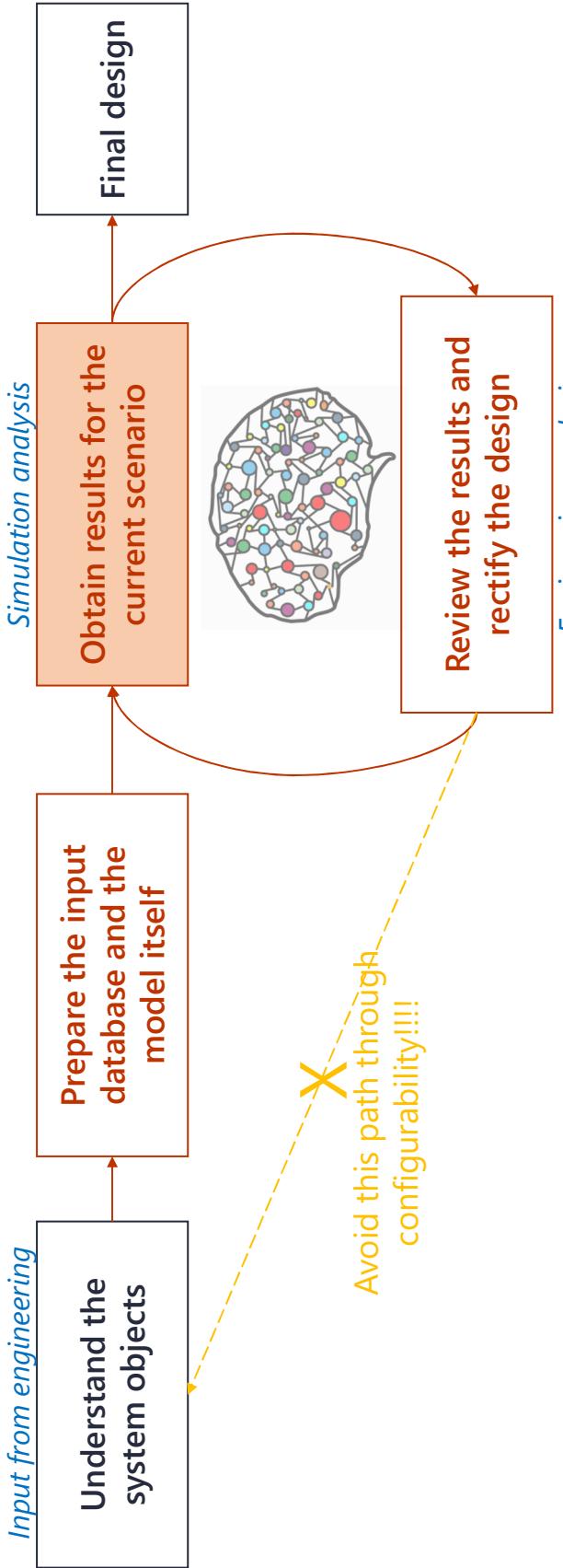
# Upfront efforts always pay!

≡ Some benefits of this approach:

- Standardized model development process
  - Opens to concurrent model editing!
  - Opens to quality management
  - **One model source file for all scenarios!**
- When iterating, adding model features is much easier
  - Not "will", but "when & what" will engineers think of down the road?
  - Adding objects and input tables is simpler with the diagrams
- Documenting trials is "built in"
  - One file or DB instance per scenario
  - "Uhhh... What did we try??" easy to answer!
- Overall less effort during the whole engineering cycle

# Iterative simulation-assisted designing

≡ Engineering iterations are much faster and easier



# A flexible and limitless platform





# A flexible and limitless platform

# Example 1: scheduling for a food processing plant



# On the complexity of dairy plants

---

≡ Why is scheduling so complicated?

- High number of possible production lots permutations
- Different and variable lot/batch sizes must be used
- Product incompatibilities need to be managed
- Allocation and priority to use share resources (CIP...)

≡ What is it all about?

- Eliminate undesired waiting times
- Shorten run lengths and avoid over-time
- Manage limited holding tanks/silos properly

# Evolving the plant...

## ≡ The weekly “ordinary” challenges

- Accommodate additional volumes: how will we do it?
- Re-scheduling during the week because...?
- Eliminate overtime and deliver on time

## ≡ Plant expansions

- How many additional silos, which capacity?
- How many new fillers, required flowrates?
- What is the effective new capacity?

## ≡ Network-wide optimization

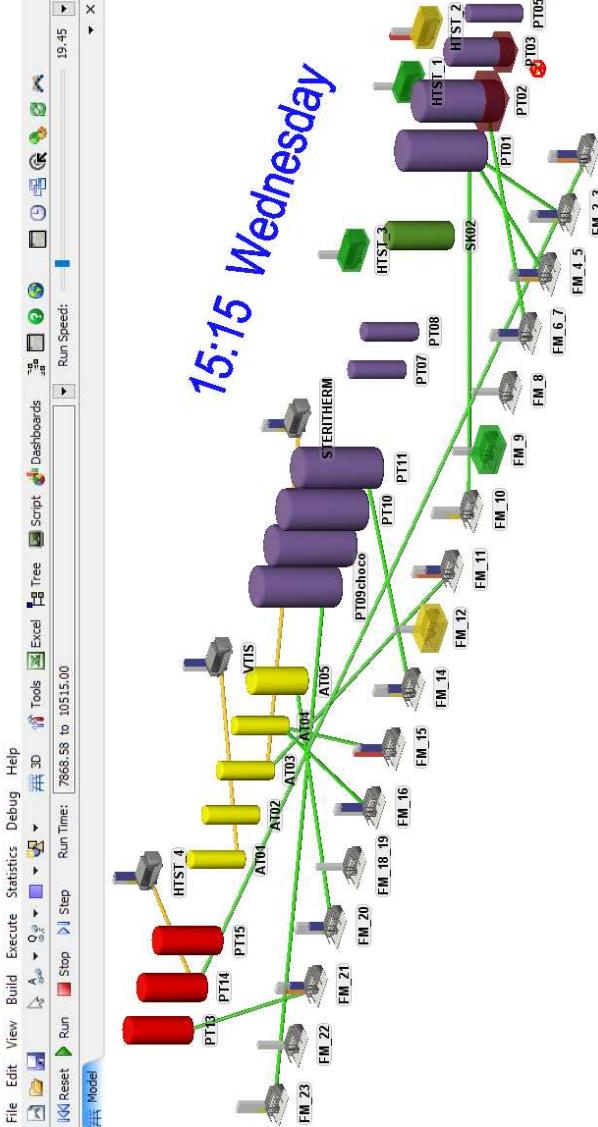
- Redistribute the total load across several plants?
- Minimize trucking to client warehouses?

# visit of the model...

≡ The input...

≡ The model...

≡ The outputs...



# Using the model

Improving an historically hand-made refined schedule:

Day:	1	2	3	4	5	6	7	GLOBAL
R1	100	100	100	50	50	90		
R2	100	100	80	80	80	96		
R3	100	100	67	67	67	91		
R4	100	100	80	100	100	100		
R5	100	100	80	100	100	100		
R6	100	100	80	43	86	96		
R8	100	100	100	50	100	100		
R9	100	100	100	100	100	100		
R20	83	100	100	100	75	100		
R21	100	100	100	100	100	100		
R23	100	100	100	100	100	100		
R24	100	100	100	100	100	100		

	Before	After	Gain
Total lost time (h)	254	60	-76%
Total available time (h)	890	1,021	+15%
Orders fulfilment	94.8%	100%	

Day:	1	2	3	4	5	6	7	GLOBAL
R1	100	100	100	100	100	100	100	100
R2	100	100	100	100	100	100	100	100
R3	100	100	100	100	100	100	100	100
R4	100	100	100	100	100	100	100	100
R5	100	100	100	100	100	100	100	100
R6	100	100	100	100	100	100	100	100
R8	100	100	100	100	100	100	100	100
R9	100	100	100	100	100	100	100	100
R20	83	100	100	100	100	100	100	100
R21	100	100	100	100	100	100	100	100
R23	100	100	100	100	100	100	100	100
R24	100	100	100	100	100	100	100	100

After 1 hour of simulation work



# Using the model

≡ Initial purpose: plant expansion

- This model was developed to support and engineering project
- The project mission was to increase by 75% the plant capacity
- The model helped sizing the new silos and deciding which new fillers and pasteurizer upgrades were the best options
  - Optimal schedules were developed at each stage of the design

≡ Then, the client asked to get the model for its own use

- Wanted to leverage this new value adding tool
- With limited training, the client reconfigured the input file to model several other plants... without the modeler!
- No need to modify the simulator's code, only Excel work...

~~Customizable~~

## Example 2: material handling and traffic debottlenecking

# Material handling in a mine

---

≡ Why is it so complicated?

- Lots of vehicles moving around for various purposes
- The scheduled tonnage must be executed on time to avoid production delays and profit margin shrinkage
- It's not an open free space... most tunnels are one vehicle wide!
- Queuing vehicles physically block tunnels... congestion!

≡ What is it all about?

- Determine where to position ore passes, truck chutes, etc.
- Determine how to smoothly mine stopes (material routing)
- For any year: how many vehicles of each type are required? What are the required passes, chutes and bins capacity?

# More challenges

---

Many parameters to adjust simultaneously:

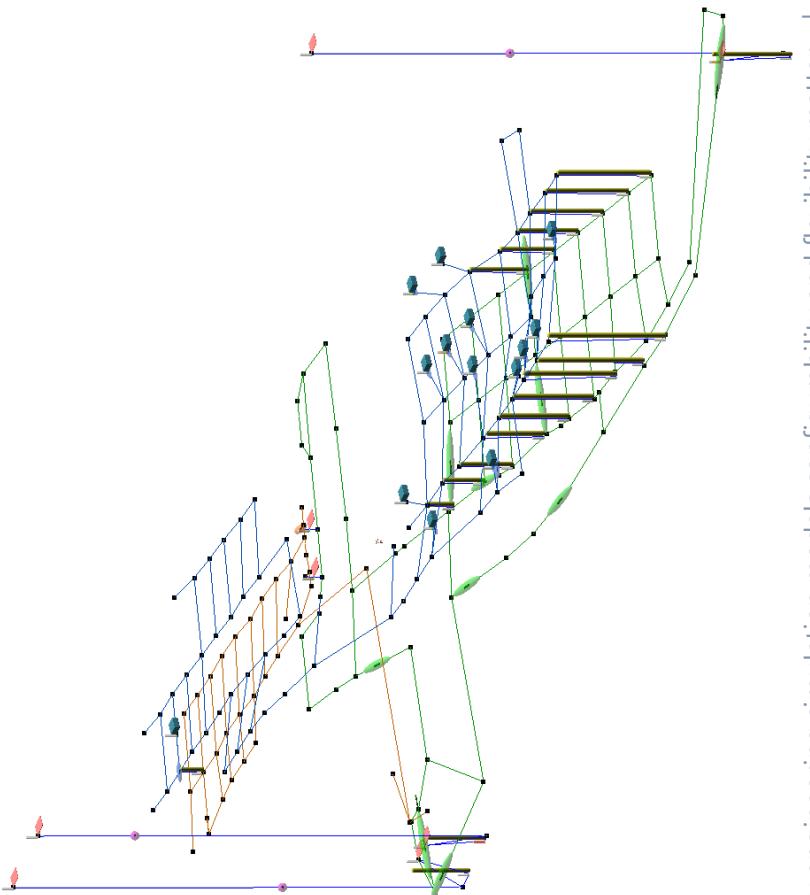
- Truck capacities and speeds
  - Number of trucks of each type
  - Priority rules to manage traffic
  - Routing rules (material going from where to where)
  - Location of intersections
  - Location of ore passes and their capacities
  - Sequencing of drifts development
  - Fuel stations location and refuelling rules (avoid queuing)
- ... for every year of the life-of-mine plan!!!

# visit of the model...

≡ The input...

≡ The model...

≡ The outputs...



# Using the model

## ≡ Example of simulation-driven "discoveries"

- Crushing rate is not fast enough because ore arrives in batches, not with a steady flow
- Coarse ore bin is too small and blocks trains
- Fine ore bin is way too big, over-design!
- The required number of active LHD to achieve the budgeted tonnage exceeds the ventilation limitation
- The refuelling schedule is inadequate: trucks wait 2 hours per day
- Additional ore passes are required closer to the stopes to reduce LHDs travel time

# How engineers had fun!

≡ During a 2-days session with the simulation:

- Crushers were installed, relocated, resized
- Ore passes were sank where they were not planned
- Different truck types were purchased and tested
- Trains switched from diesel to electric power
- Second shaft sinking was executed 5 years earlier than planned
- Operators were hired to replace drivers during lunch
- Crusher control room was moved to surface
- ....

~~Customizable~~

# Final words



d

# If you ask yourself...

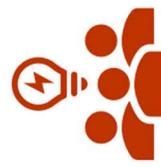
- ≡ Can the system attain its designed throughput?
- ≡ How can I scale up my production capacity?
- ≡ Are conveying and/or storage capacities adequate?
- ≡ Are there sufficient cranes or vehicles?
- ≡ Can the supply chain respond to client requirements?
- ≡ Is the schedule of operations feasible?
- ≡ What will be the impact of breakdowns on throughput?
- ≡ What are the optimal spare parts inventory levels?
- ≡ ..

Think  
Simulation!



# In conclusion...

- ≡ Discrete Events Simulation is a powerful technique to model flow of items, schedules, operators tasks, and interactions between system elements
  - The exercise of building the model in itself brings to daylight **foreseeable** design flaws and operational inefficiencies
  - Simulation results helps to assess the impact of changes before really changing the system
  - A highly configurable model is a powerful and valuable support to engineering designs validation & optimization
  - Team consensus is built around an analytical tool...
- ≡ **Simulation is beneficial to select solutions having a global impact!**





**Difference** is a society offering coaching, consulting and training services in statistic, data science, simulation and continuous improvement.

We promote the use of quantitative tools that can be applied at the different steps of an improvement and variability reduction project.



Powerful  
methods



Adapted  
approach



Combining harc  
work with fun

For more information, you can contact:

**Martin Carignan**, M.Sc., MBA  
Principal Associate  
+1 (514) 795-8000  
[mcarignan@difference-gcs.com](mailto:mcarignan@difference-gcs.com)  
[linkedin.com/in/martincarignan](https://linkedin.com/in/martincarignan)

**Vincent Béchard**, MASC.  
Associate, Analytical Decision Specialist  
+1 (438) 521-5829  
[vbechard@difference-gcs.com](mailto:vbechard@difference-gcs.com)  
[linkedin.com/in/vincentbechard](https://linkedin.com/in/vincentbechard)