# CODE-FIRST PROCESS MODELING AND ANALYSIS WITH KALASIM

## UNDERSTAND AND OPTIMIZE REAL-WORLD PROCESSES AT EASE




International Winter School on Software Engineering

Dr. Holger Brandl

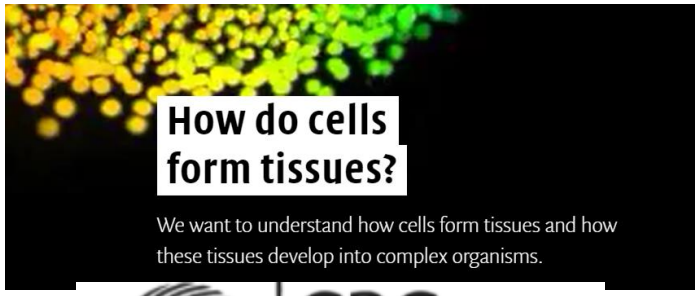
# OUTLINE

- Introduction
- Process Analysis
- Discrete Event Simulation
- Kalasim Basics
- Simulation Analysis
- Exercise 1
- Process Animation
- Process Optimization
- Exercise 2
- Summary & Next Steps

# ABOUT ME


HONDA

Universität Bielefeld




CBG
Max Planck Institute of Molecular Cell Biology and Genetics

SYSTEMA
Art of Automation

INDUSTRY 4.0


2010 — 2015 — 2020

GitHub Hobbies

`kscript` `krangl` `kravis` `kalasim`

systema.com/optimization

Blog Resources Support Contact

SYSTEMA
Art of Automation

INDUSTRY 4.0 COMPANY INDUSTRIES CAREER

Digital Transformation

Smart Manufacturing

SYSTEMA Portfolio

SAP Portfolio

Resources

Integration

Automation

Optimization

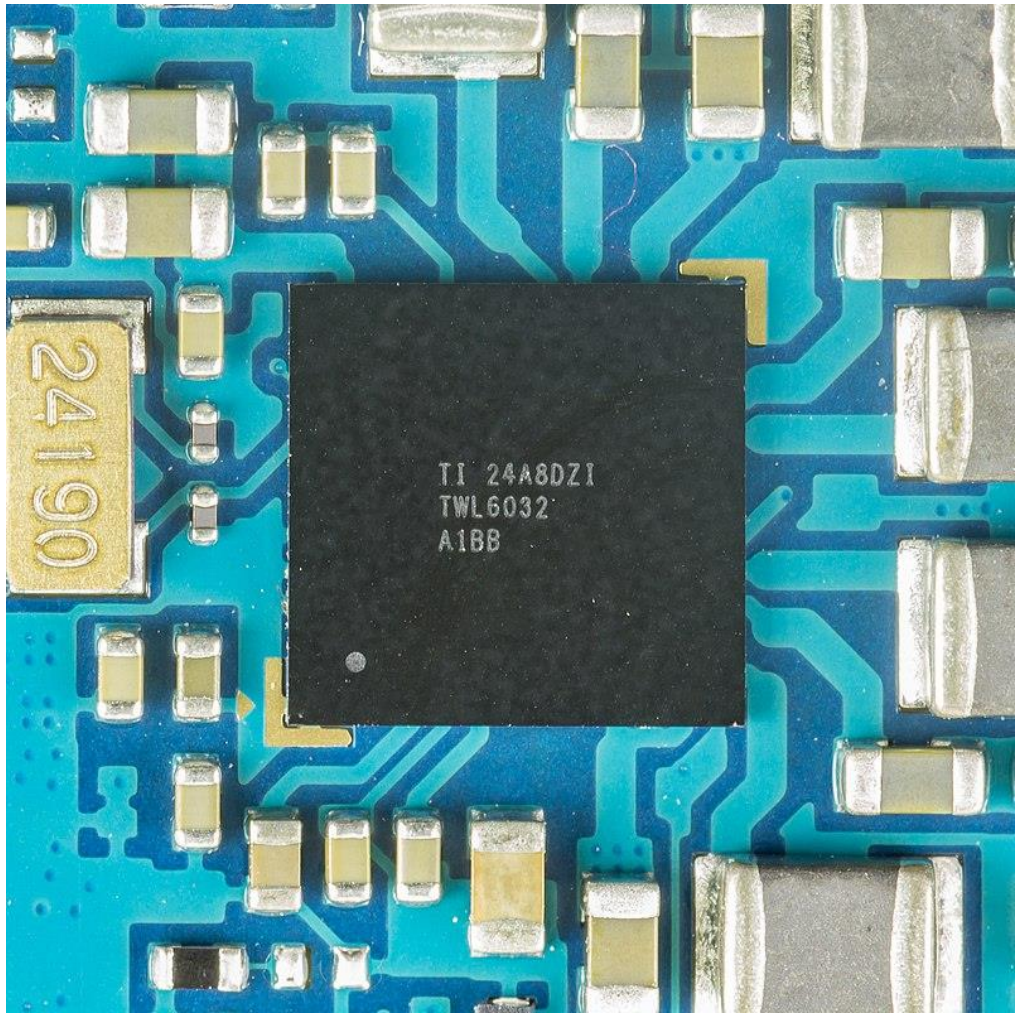
Visualization

Migration

Optimization is the art of maximizing manufacturing efficiency, throughput, OEE, yield, and quality by monitoring, analyzing, and iteratively tuning manufacturing processes.

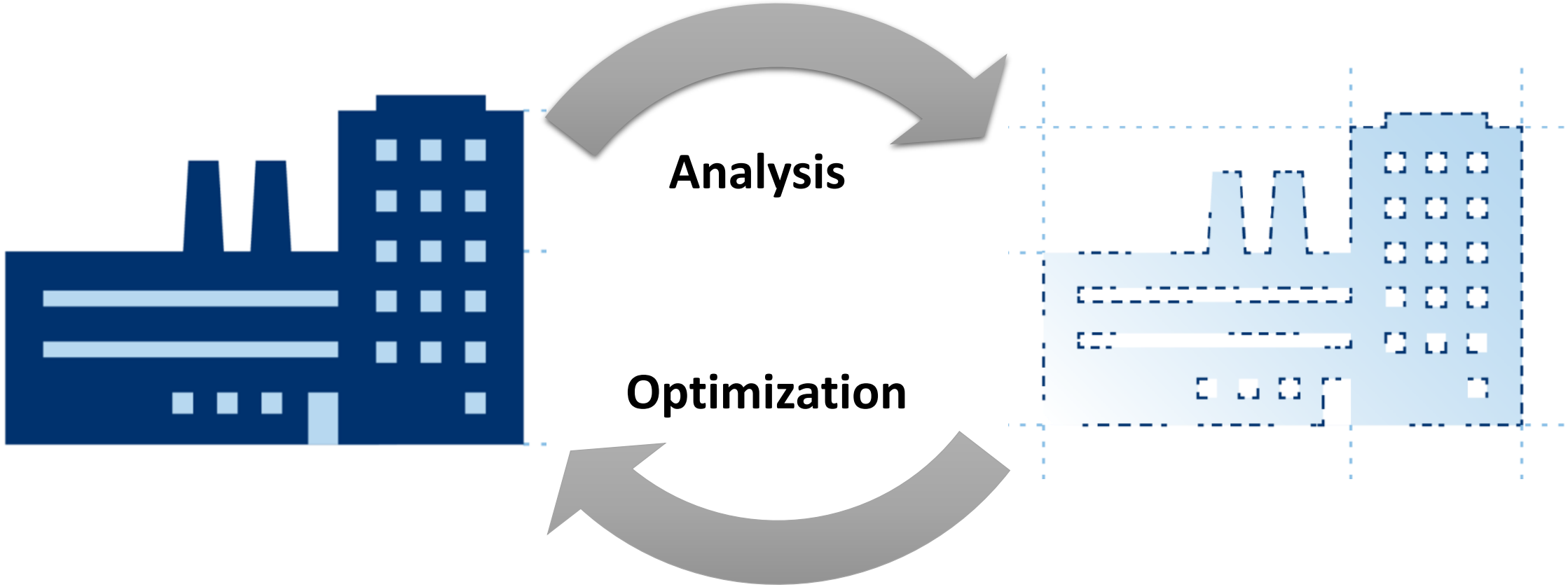
# CHALLENGE: SEMICONDUCTOR PRODUCTION

- Cycle Time: ~ 6 Months
- > 1000 Steps
- Product routes reiterate processes to add more layers on top of raw silicon wafer
- Microprocessors are the most complex manufactured product on earth
- 30 floor “sky” scraper built with nanometer precision

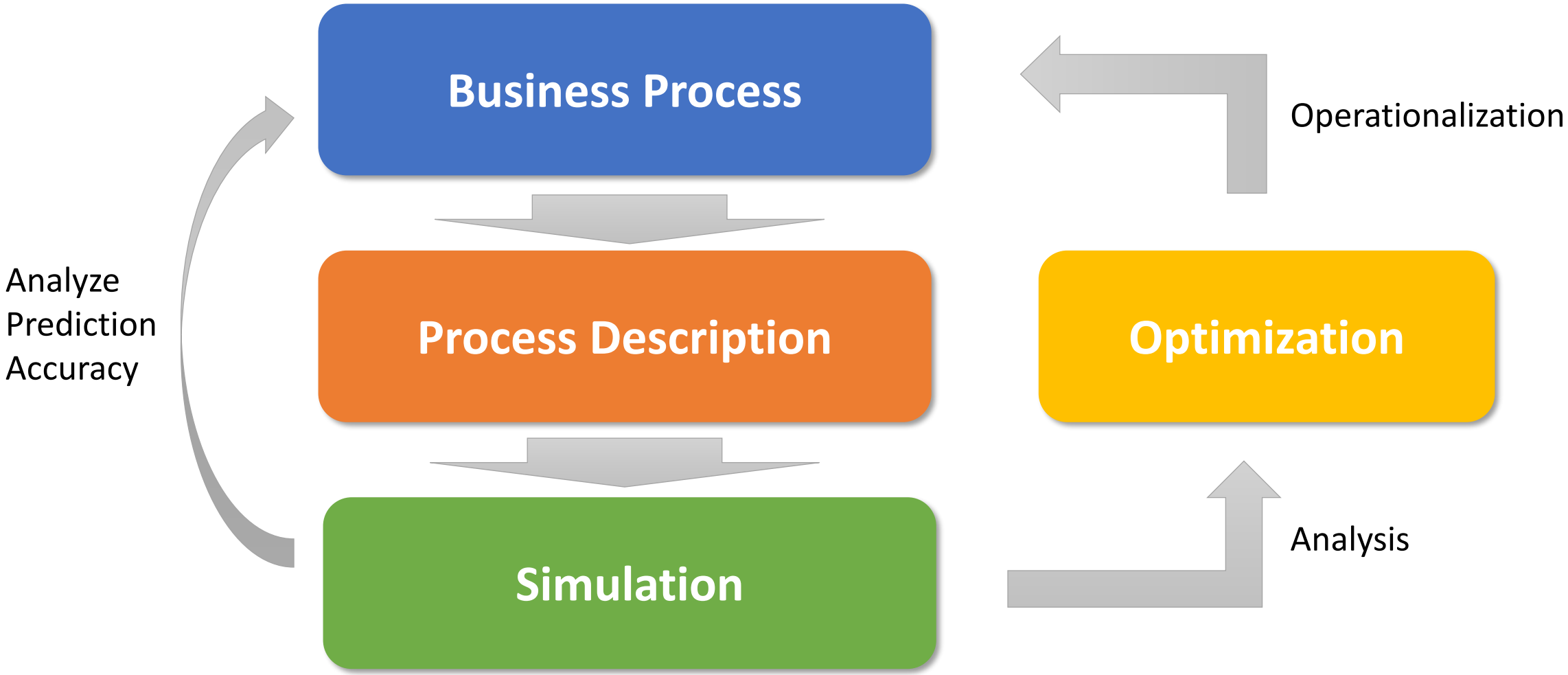



HOW TO MAKE MORE MONEY?

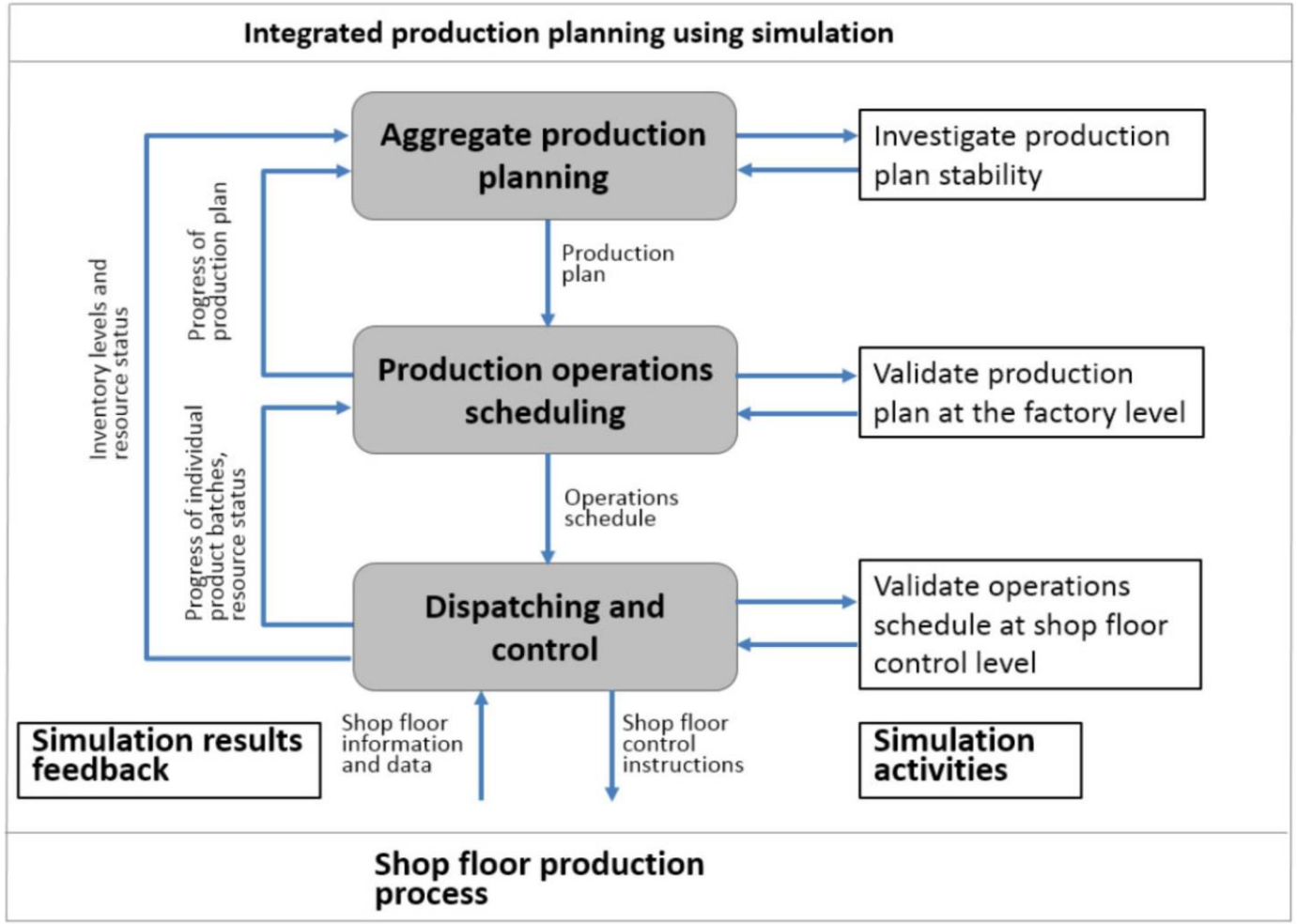
# HOW TO OPTIMIZE A COMPLEX SYSTEM?

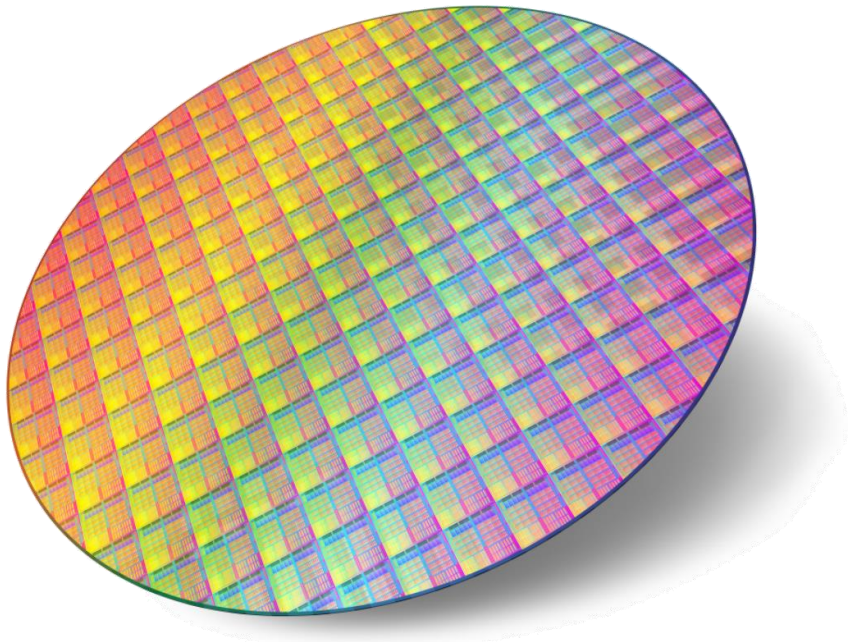


# PROCESS OPTIMIZATION LIFECYCLE



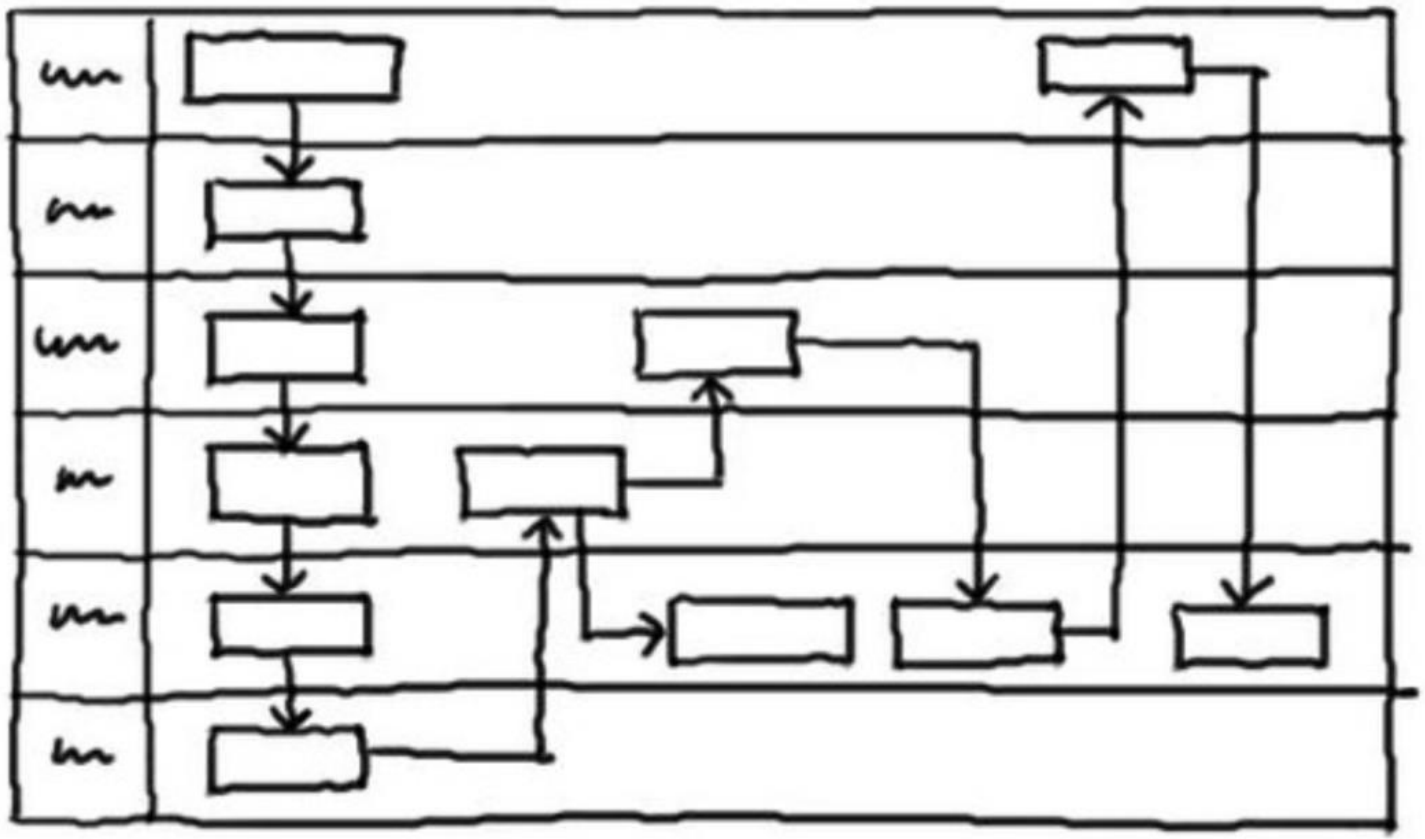
# HOW TO PRODUCE MORE IN COMPLEX INDUSTRIES?



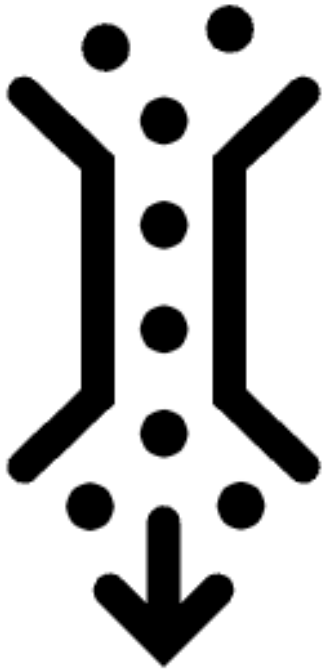
Kibira et al., WSC-2016

# MODEL THE BUSINESS PROCESS!



NICE, BUT WHERE IS THE BOTTLENECK?

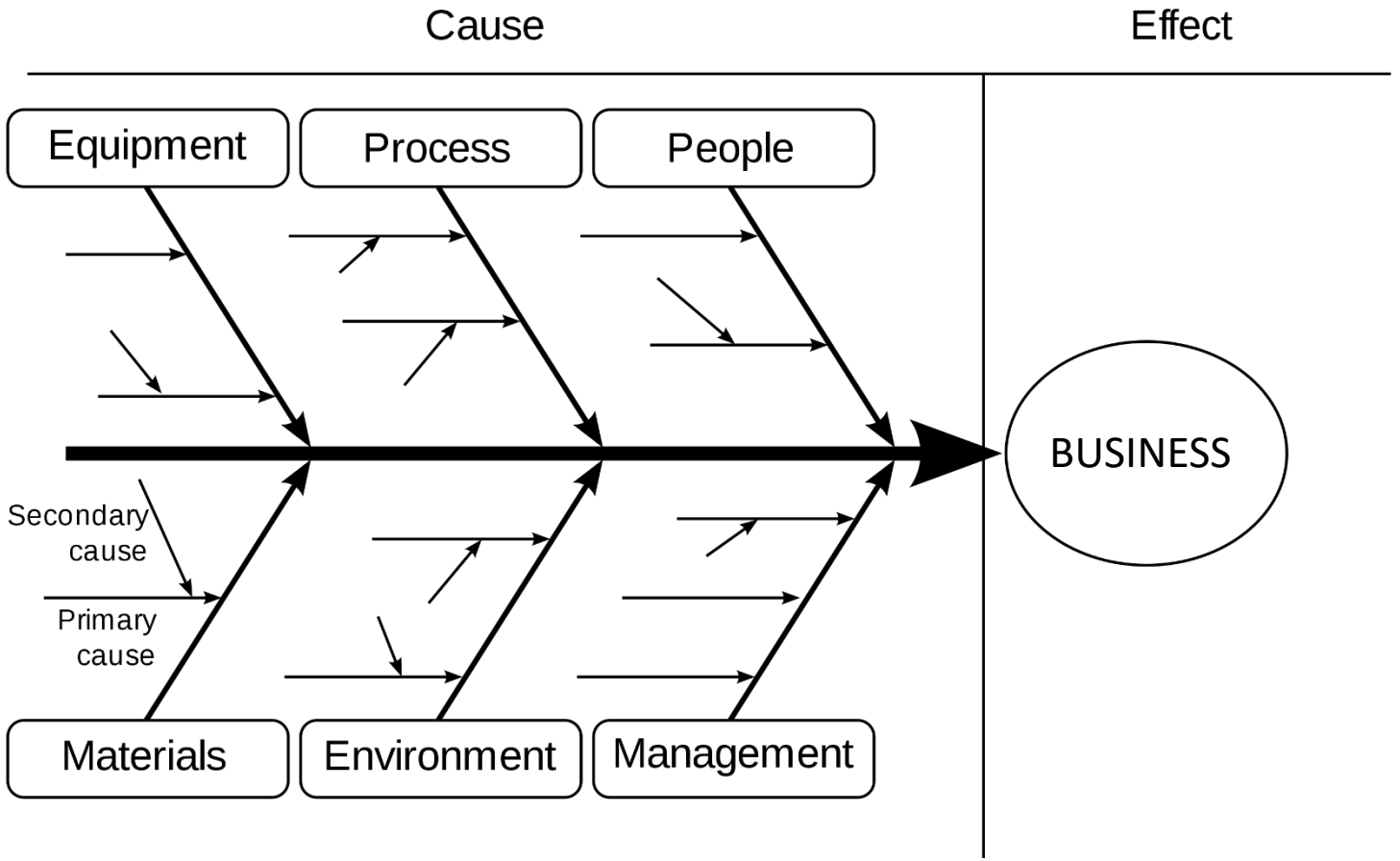
# WHERE IS THE BOTTLENECK IN MY BUSINESS?



- A bottleneck is a process  where one process reduces capacity of the whole chain of processes
- A bottleneck worsens or delays the process or its performance, increasing the cycle time or even bringing the process to a complete standstill. Thus, a bottleneck usually holds a lot of potential for optimization. Bottleneck sources can be found and eliminated by Process Discovery and root cause analyses.

https://lanalabs.com/en/glossary/bottleneck

# CAUSE AND CONSEQUENCE

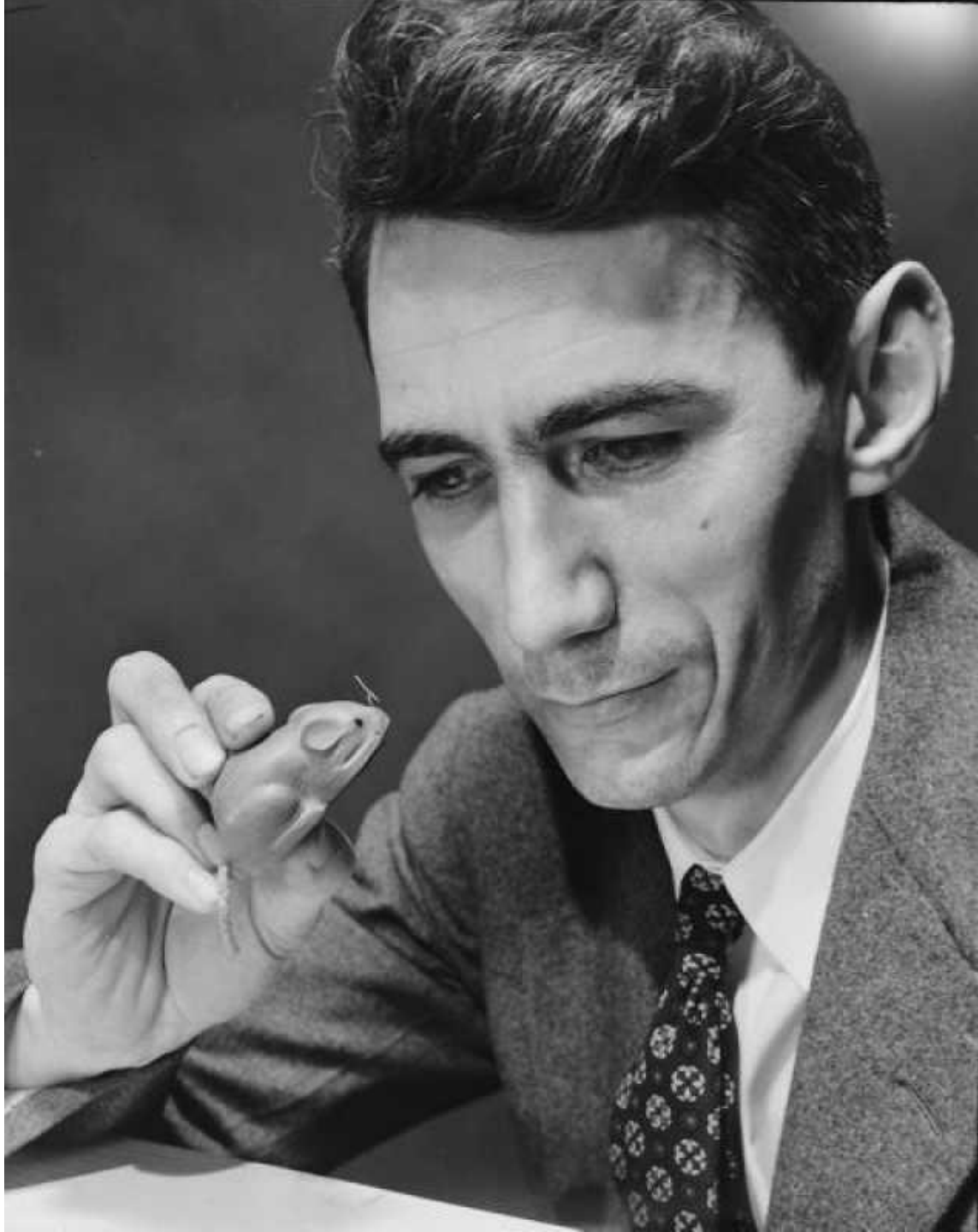



- Complex interplay of human, machine, material and methods
- Not always following intuition
- Applies to all businesses from service industry to manufacturing

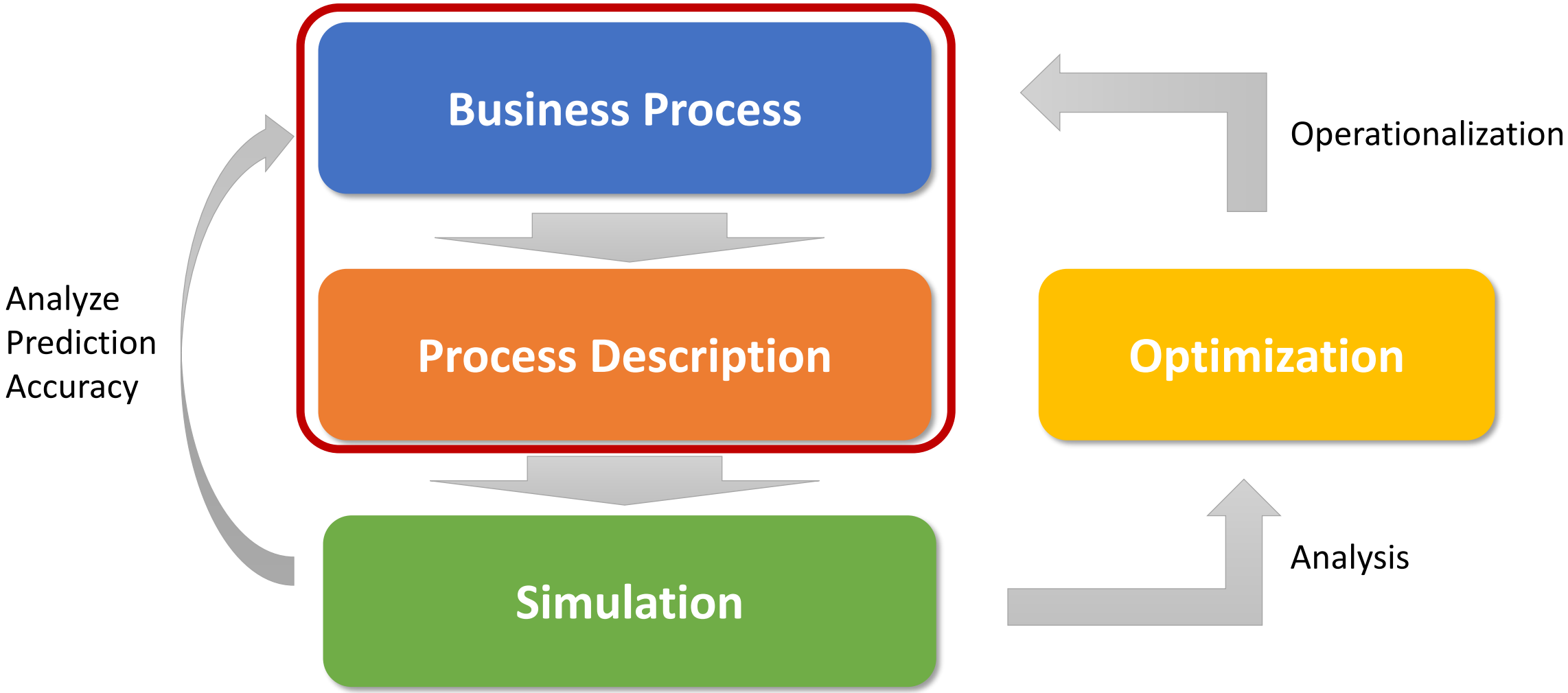
# WHAT IS SIMULATION?

A simulation is the process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the behavior of the system or of evaluating various strategies (within the limits imposed by a criterion or a set of criteria) for the operation of the system.

Claude Shannon (1975)



# UNDERSTAND YOUR PROCESS



# DEEP DIVE INTO THE PROCESS

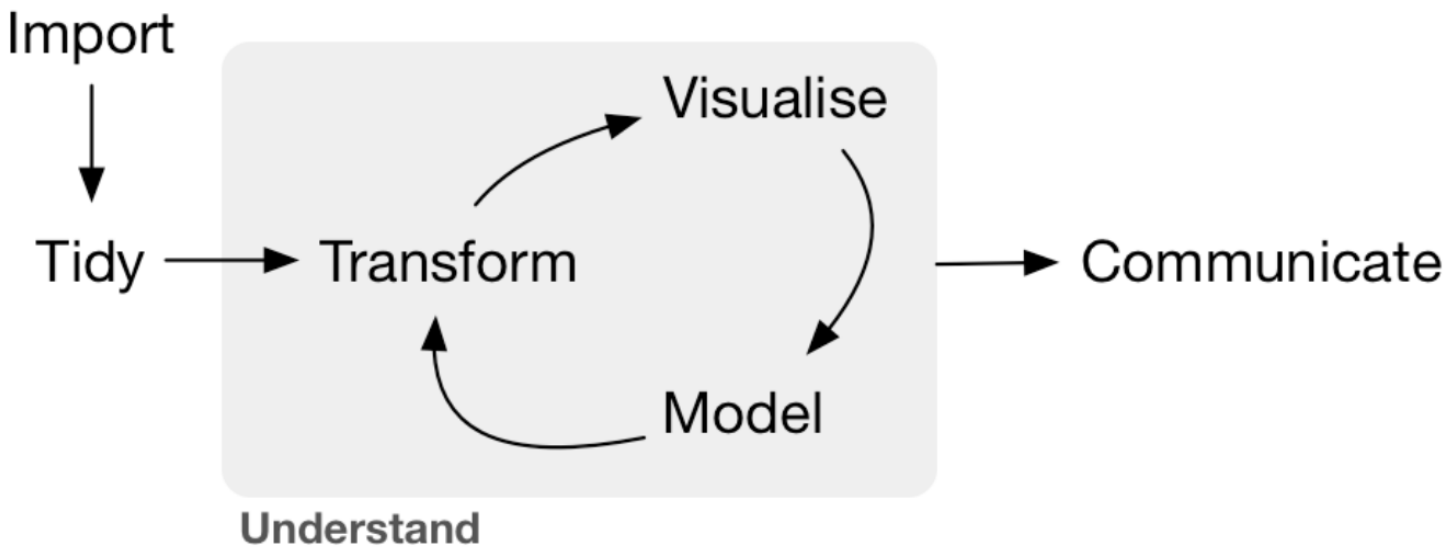
Real-world processes are complex & sometimes seemingly unstructured

1. Work out **process specification**
- Processes
- Actors
- Interactions
- Data sources
- Data gaps
- Formalize control regimen

2. **Analyze data** from the process

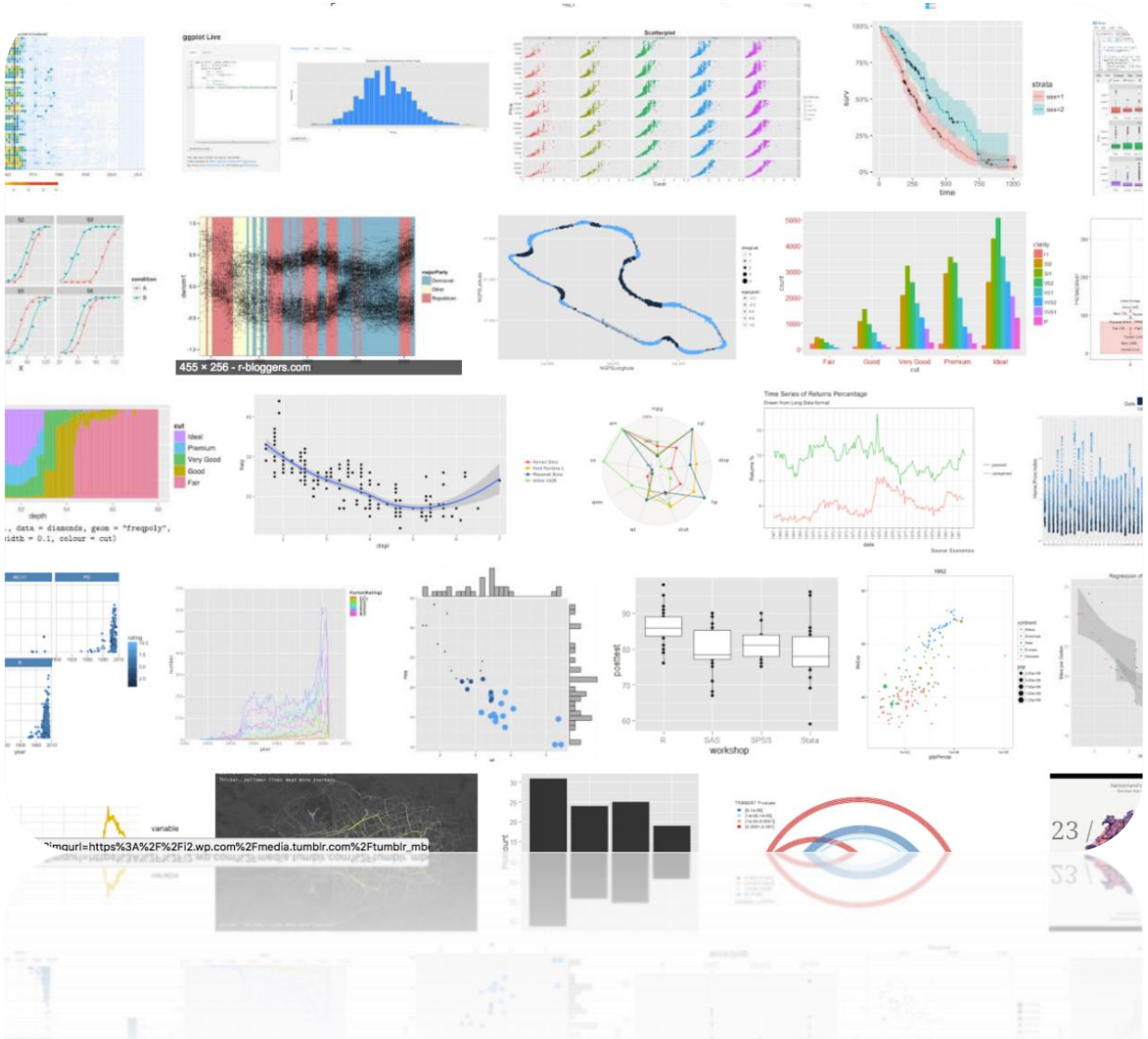
Iterative process with stake-holders

# THE DATA-SCIENCE LIFE CYCLE




- Similar for process modelling & analytics
- Model → Simulation
- Import → Process Description & Data Snapshots

# PROCESS ANALYSIS → DATA ANALYSIS




- R, Python or similar
- Work out means, distributions
- Assess for variations in time
- Association & Correlation analysis
- Statistical analysis of relevant process parameters

# ANALYSIS: HOW TO?




# A modern programming language that makes developers happier.

Get started | Why Kotlin

Developed by JetBrains & Open-source Contributors

### Multiplatform Mobile

Share the logic of your Android and iOS apps while keeping UX native

### Server-side

Modern development experience with familiar JVM technology

### Web Frontend

Extend your projects to web

### Android

Recommended by Google for building Android apps

Multiplatform for Other Platforms, Data Science

# DATA SCIENCE WITH `kotlin`?

Tutorials

- https://kotlinlang.org/docs/data-science-overview.html
- http://holgerbrandl.github.io/krangl/10_minutes/
- http://holgerbrandl.github.io/krangl/data_manip/

Talks

- KotlinConf2017 - A wild ride through New York City with Holger Brandl
- KotlinNight2018 - Kotlin's emerging data-science ecosystem with Holger Brandl
- KotlinConf 2019 Making Kotlin Ready for Data Science by Roman Belov


Social

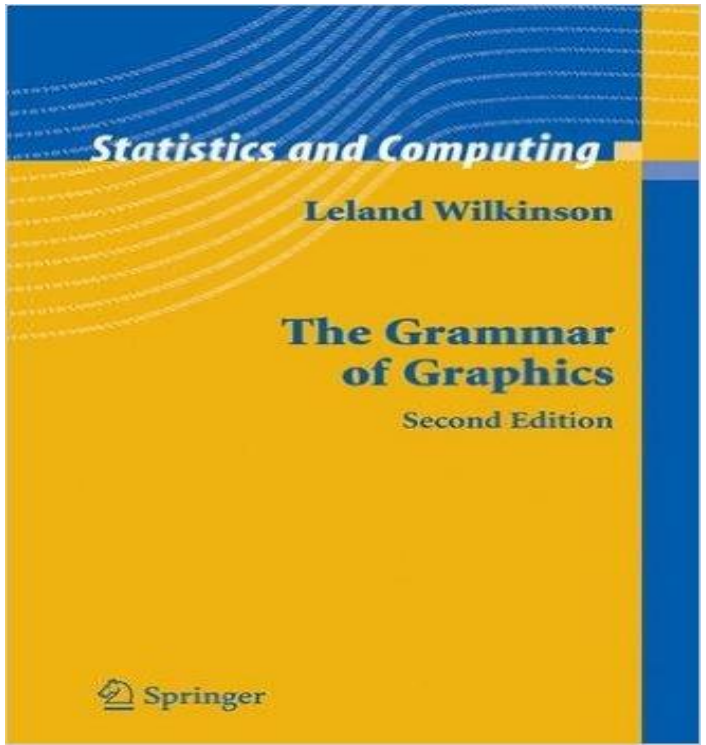
- Kotlin Slack channels science and data-science & github

# QUO VADIS, `kotlin` FOR DATA SCIENCE?

- Amazing language  for modelling & prototyping
- Great projects dataframe, KotlinDL or multik
- Awesome jupyter kernel
- Tooling barrier easy to climb for developers but harder for data scientists
- Slow data-science community adoption
- Multiplatform & Huge JVM ecosystem

# WHY R FOR PROCESS DATA ANALYTICS?

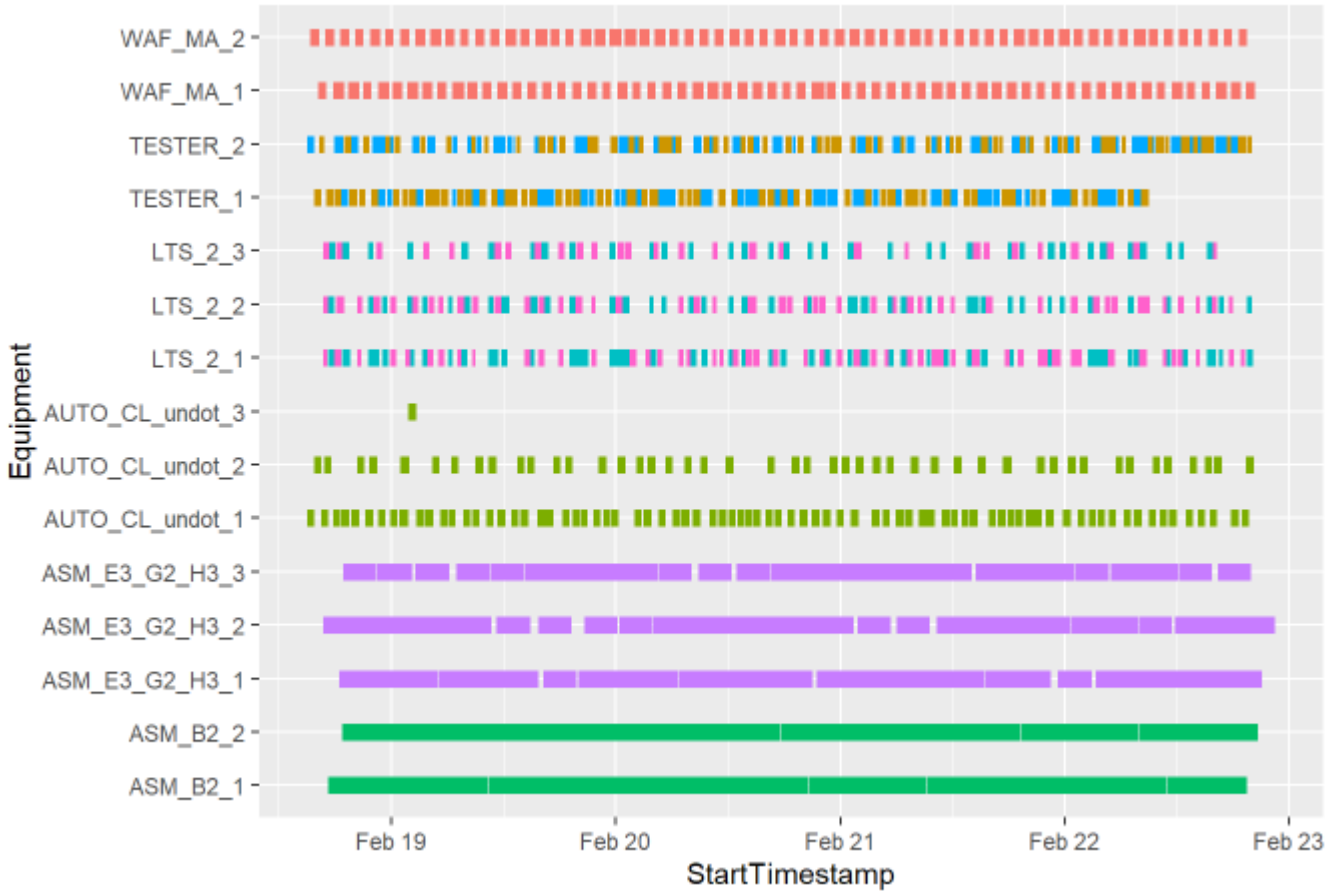
1. Superior table manipulation with `dplyr`
2. Superior data visualization with `ggplot2`
3. Superior statistics toolbox



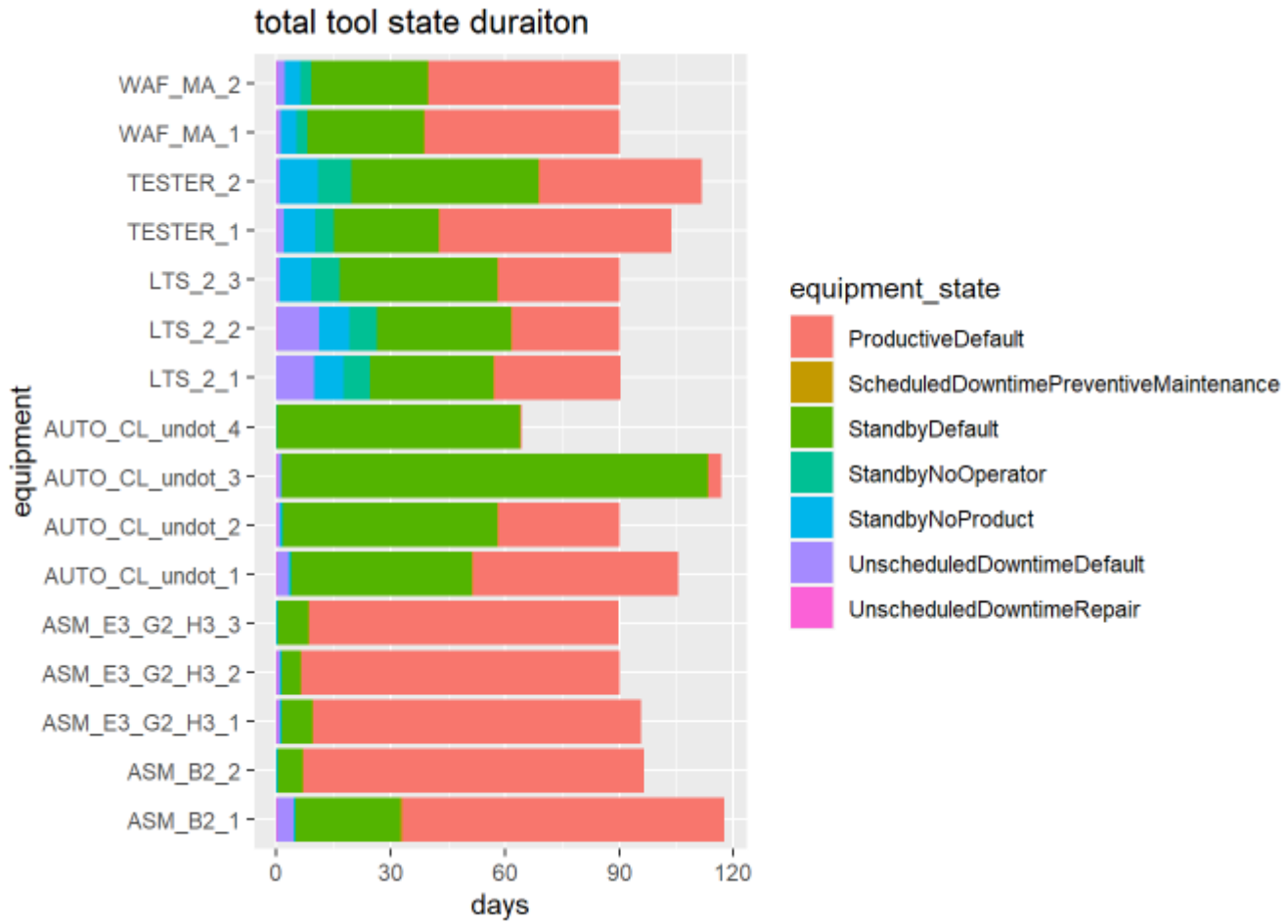

Statistics is about proving what you expect, while visualization is about discovering what you didn't expect.

- Jan Aerts

# EXAMPLE: EQUIPMENT UTILIZATION

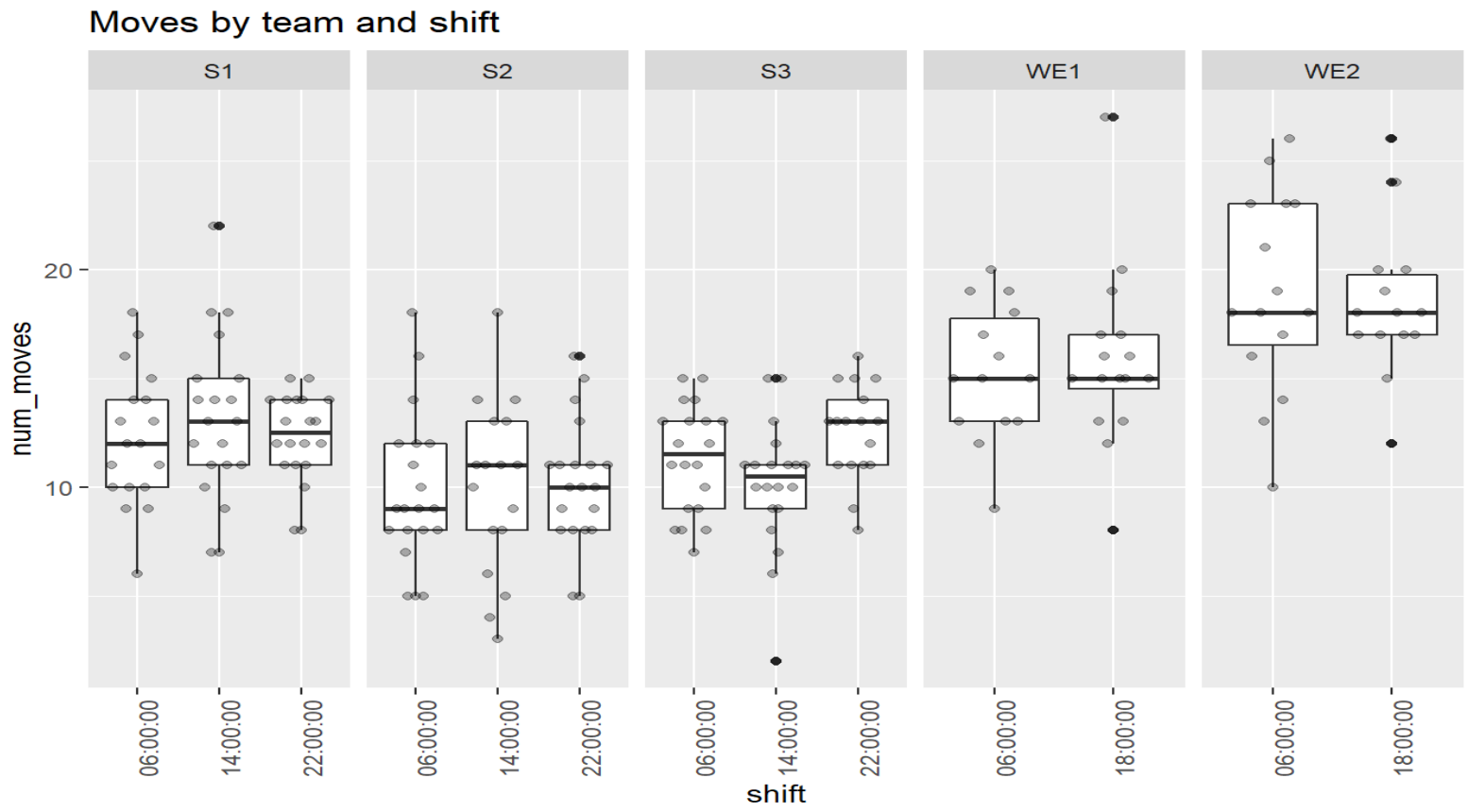


# EXAMPLE: EQUIPMENT UTILIZATION (CONT)



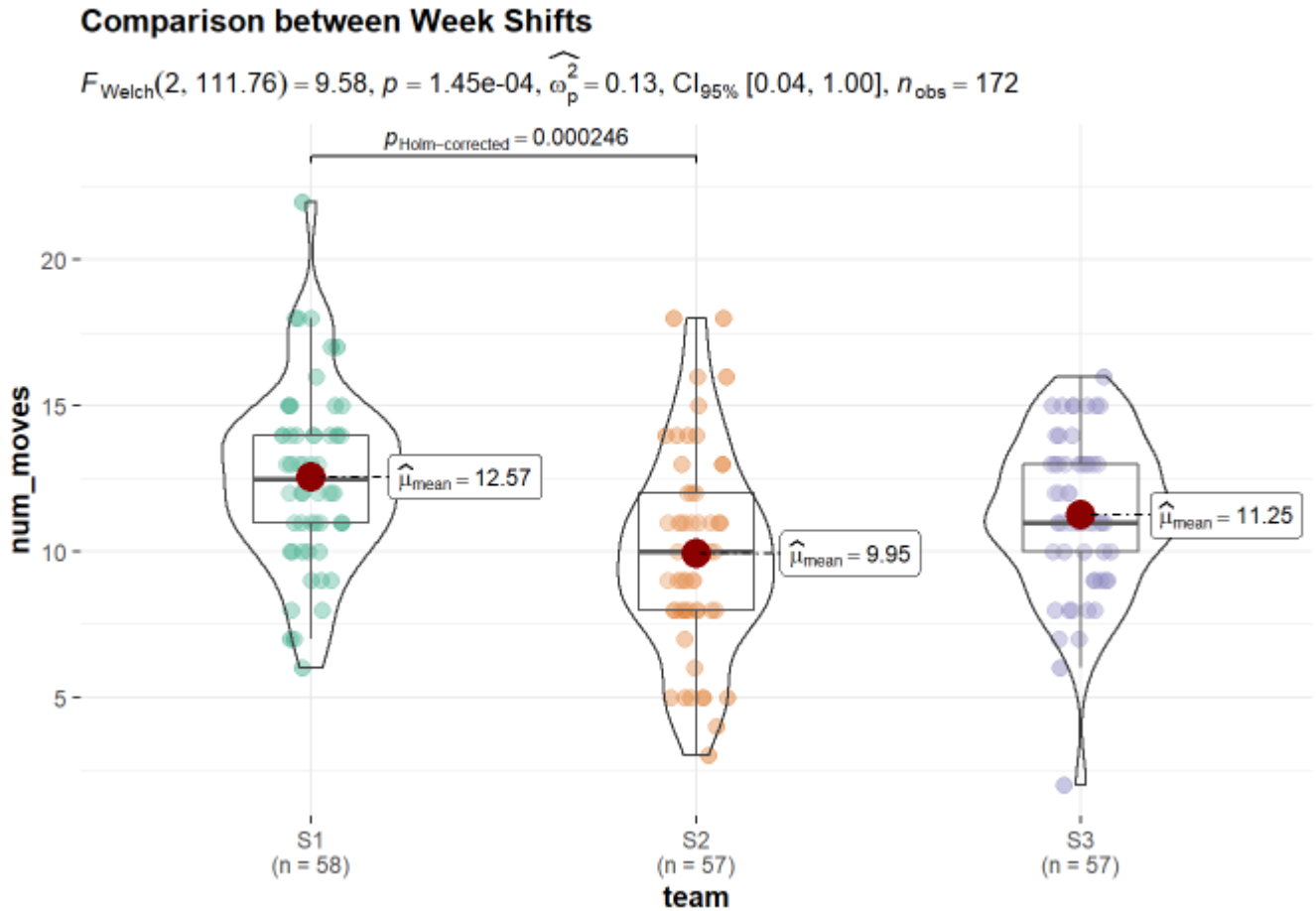
# EXAMPLE: PROCESS METRICS TIMESERIES

# EXAMPLE: SHIFT PERFORMANCE VISUALIZATION



https://github.com/IndrajeetPatil/ggstatsplot

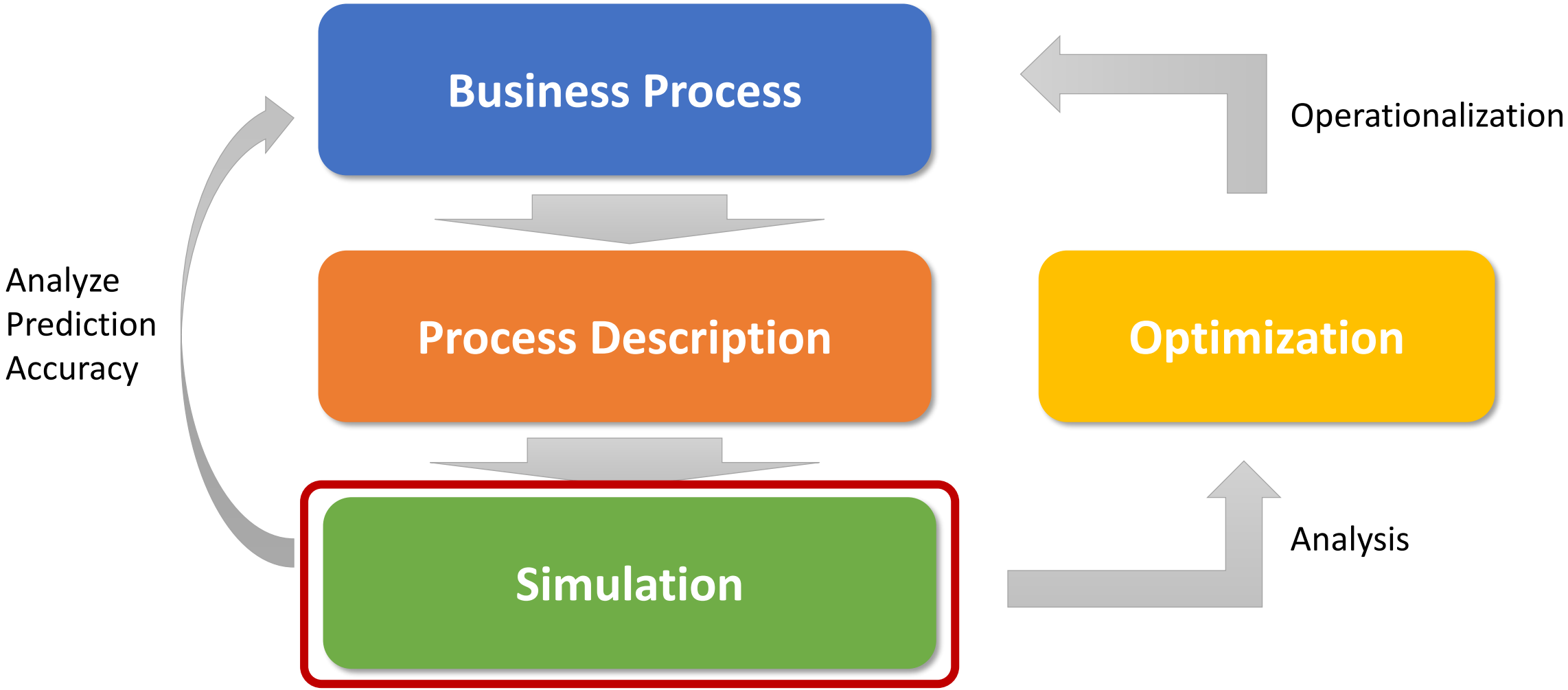
# EXAMPLE: SHIFT PERFORMANCE ANALYSIS



# PROCESS ANALYSIS FROM THE TRENCHES

- Do not trust the bosses, ask the workers
- Process description is the most important document in simulation optimization project → Requires regular review & a lot of ❤️
- Work out the process definitions from different perspective → different stories
- Always question & validate documented processes using data analysis
- Plan enough time to understand a process, you will need it

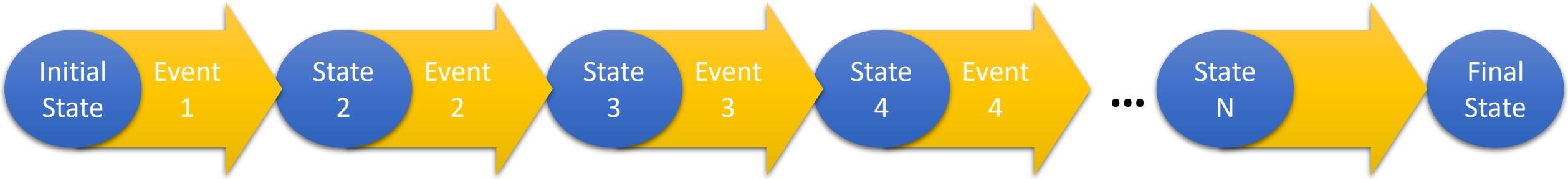
# WHAT IS SIMULATION AND HOW TO "DO" IT?



# DISCRETE EVENT SIMULATION

A discrete event simulation (DES) is a tool that allows studying the dynamic behavior of stochastic, dynamic and discretely evolving systems such as

- Factories
- Ports & Airports
- Traffic
- Supply chains & Logistics
- Controlling



# TYPICAL APPLICATIONS OF DISCRETE EVENT SIMULATION

Production planning (such as bottleneck analysis)

Dimensioning (How many drivers are needed? Number of servers?)

Process automation & visualization

Capacity Planning

Project management

# DES FLAVORS

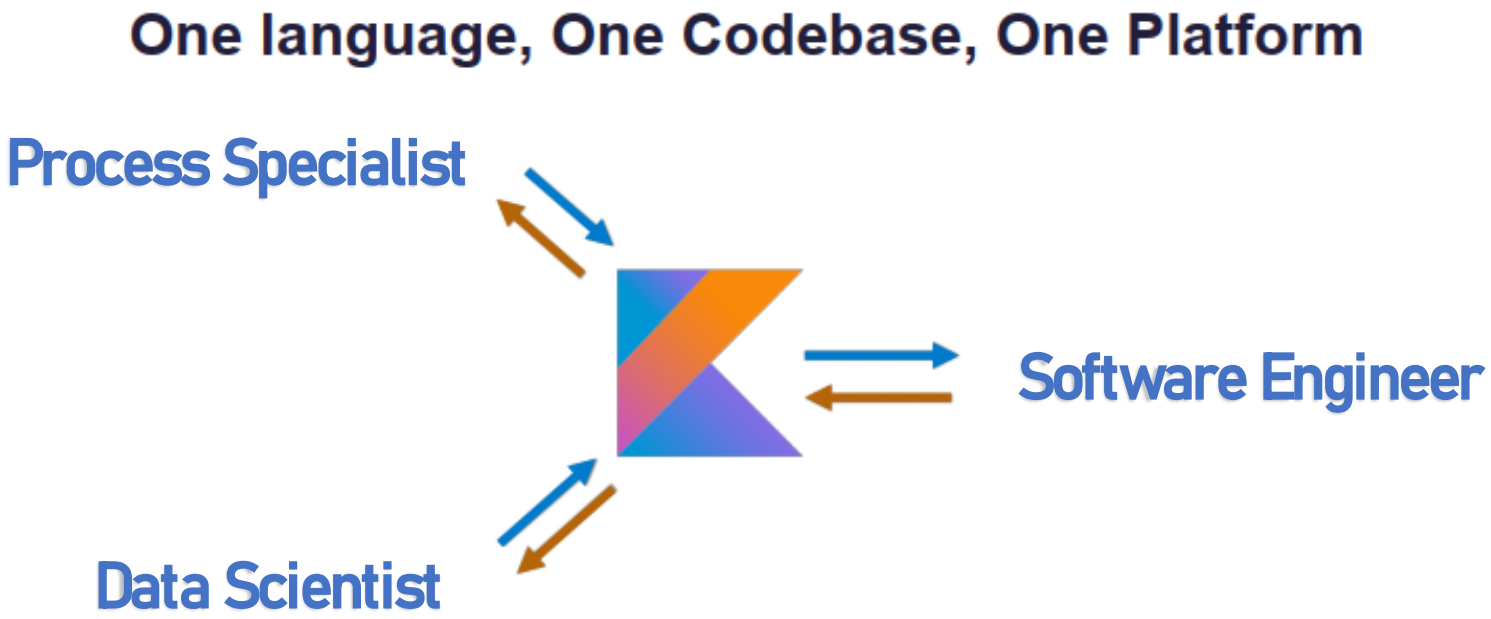
- In *activity-oriented* DES the simulation clock advances in **fixed time increments** and all simulation entities are scanned and possibly reevaluated. Clearly, simulation performance degrades quickly with smaller increments and increasingly complex models.

- In *event-oriented* DES is built around a list of **scheduled events ordered by future execution time**. During simulation, the these events are processed sequentially to update the state of the model.

- Finally, *process-oriented* DES refines the event-oriented approach by defining a **vocabulary of interactions** to describe the interplay between simulation entities. This vocabulary is used by the modeler to define the component life-cycle processes of each simulation entity.

# WHICH DISCRETE SIMULATION ENGINE?

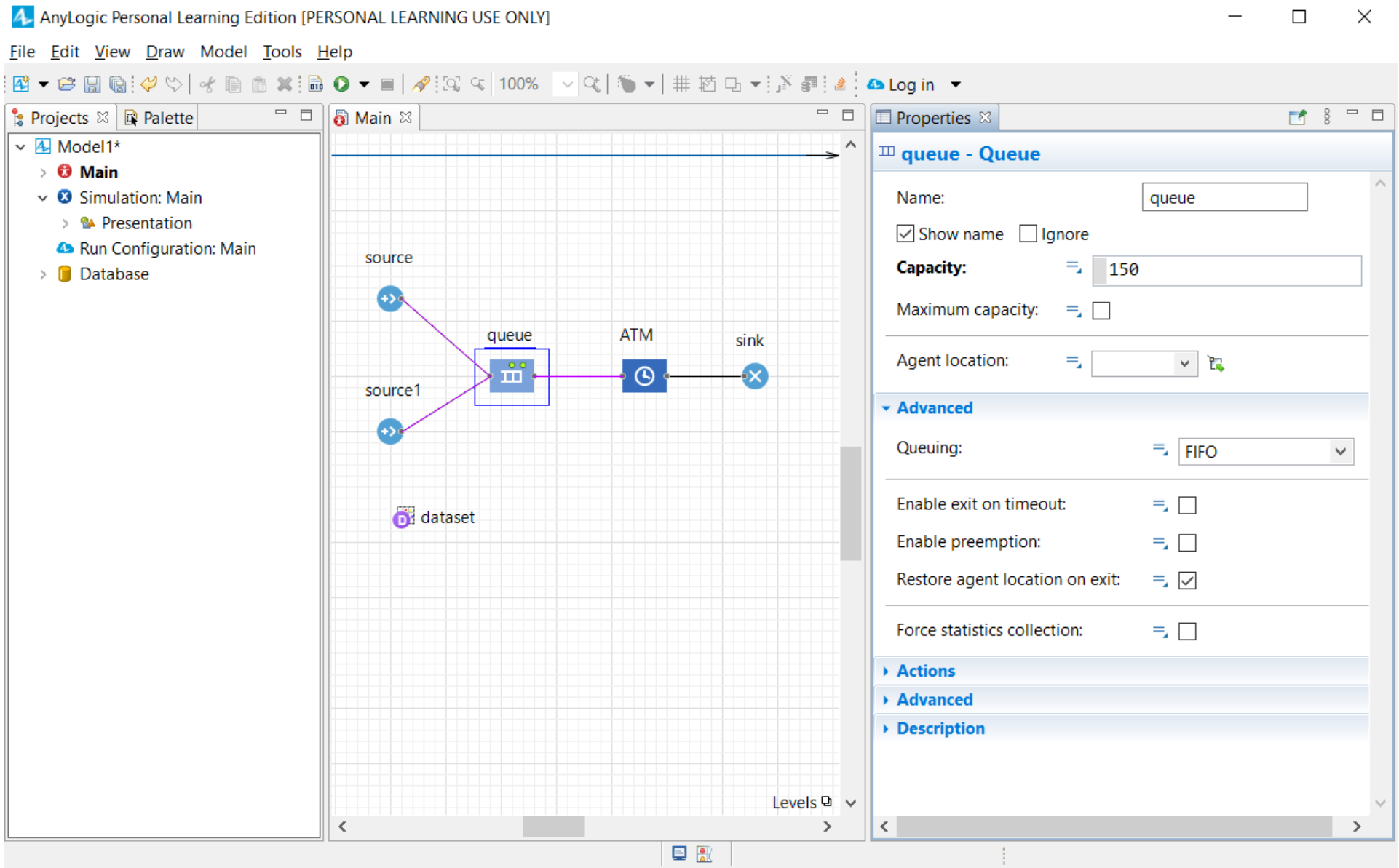
- Great existing ecosystem including Simmer, Salabim, SimJulia, SimPy, DSOL, …
- Low-code commercial tooling such as  AnyLogic or SIMIO



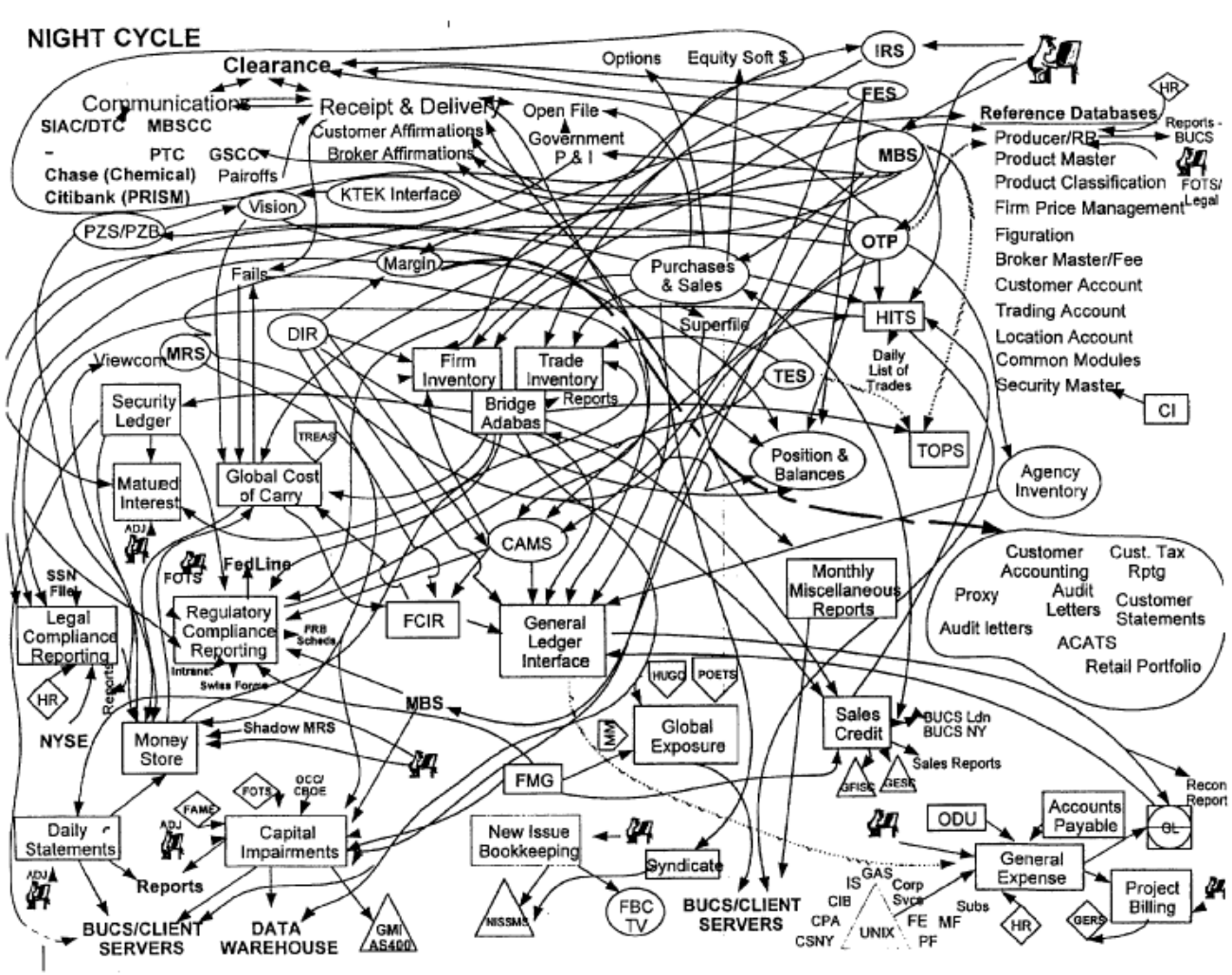
# LET'S KEEP IT SIMPLE, GO NO-CODE ?!

http://www.anylogic.com

# NO-CODE SIMULATION



# THE REAL LIFE - GOOD LUCK WITH NO-CODE

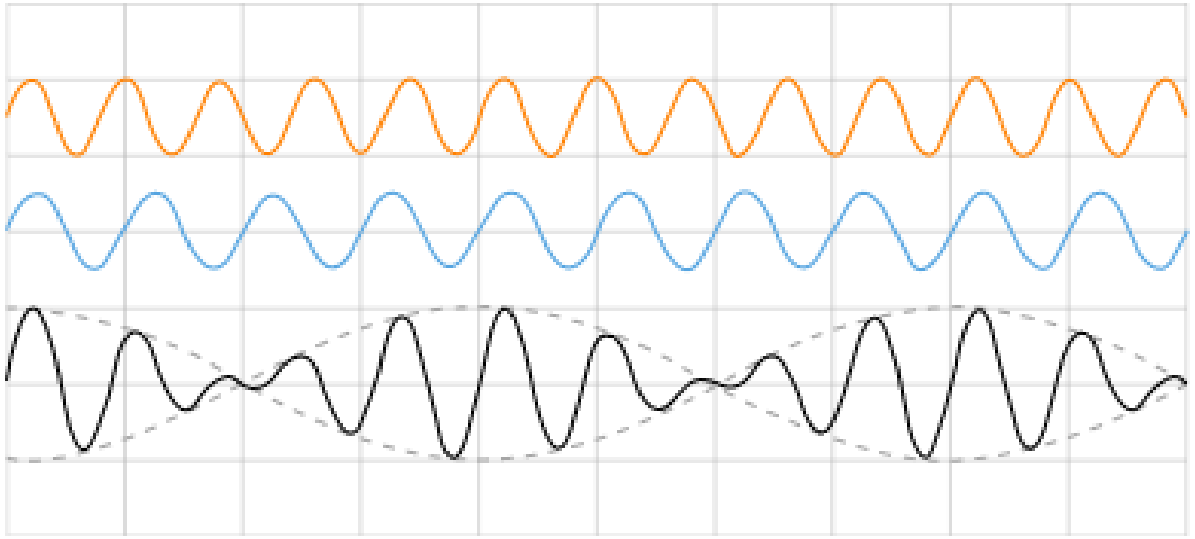



- No-code sim may work for some (simple!) applications
- To manage more complex models we need change-tracking, scaling, refactoring abilities, CI/CD, unit-tests, and the rest of the gang that makes simulation development fun.
- No-code tools typically lack some of these requirements

Source

# PROCESS SUPERPOSITION

- Similar to physics
- Many simple business processes are coevolving & interacting with each other
- Results process dynamics are hard to describe classical calculus tools
- Simulation: Build up complex model by superimposing more atomic processes



# Welcome to `kalasim`


`kalasim` is a discrete event simulator. It provides a statically typed API, dependency injection, modern persistence, structured logging and automation capabilities.

kalasim is designed for simulation practitioners, process analysts and industrial engineers, who need to go beyond the limitations of existing simulation tools to model and optimize their business-critical use-cases.

In contrast to many other simulation tools, `kalasim` is neither low-code nor no-code. It is *code-first* to enable change tracking, scaling, refactoring, CI/CD, unit-tests, and the rest of the gang that makes simulation development fun.

**Table of contents**

# holgerbrandl / kalasim

Public

Unpin | Unwatch 1 | Fork 5 | Starred 21

Code | Issues 13 | Pull requests | Discussions | Actions | Projects | Security | Insights | Settings

master | 4 branches | 31 tags | Go to file | Add file | Code

holgerbrandl Continued call-center article — e505375 23 hours ago — 673 commits

| | | |
|---|---|---|
| .github/workflows | Improved documentation | 12 months ago |
| .idea/codeStyles | Fixed #27: More informative resource events | 17 days ago |
| docs | Continued call-center article | 23 hours ago |
| gradle/wrapper | Matched letsplot display support API as much as possible | 2 months ago |
| modules | Continued elevator animation | yesterday |
| simulations | Fixed naming | 6 days ago |
| src | Continued call-center article | 23 hours ago |
| .gitattributes | override github language stats | 17 days ago |
| .gitignore | share codestyle in repo | last month |
| CHANGES.md | Improved documentation | 12 months ago |

## About

Discrete Event Simulator

data-science | simulation | optimization | agent-based-modeling | process-modeling | visulization | discrete-event-simulation

Readme

MIT License

21 stars

1 watching

5 forks

## Releases 29

v0.7.95 Latest

yesterday

+ 28 releases

# CORE FEATURES

`kalasim` is a generic process-oriented discrete event simulation engine

- Simulation entities have a generative **process description** that defines the interplay with other entities
- There is a well-defined rich process **interaction vocabulary**
- An **event trigger queue** maintains future action triggers and acts as sole driver to progress simulation state
- Built-in **monitoring and statistics** gathering across the entire API

# KALASIM APPLICATIONS

- Optimization
- AI Research
- Bottleneck Analytics
- Integration Testing
- Games & Fun
- Capacity Planning

# FIRST EXAMPLE

Key Types

- `Component`
- `Resource`
- `State`

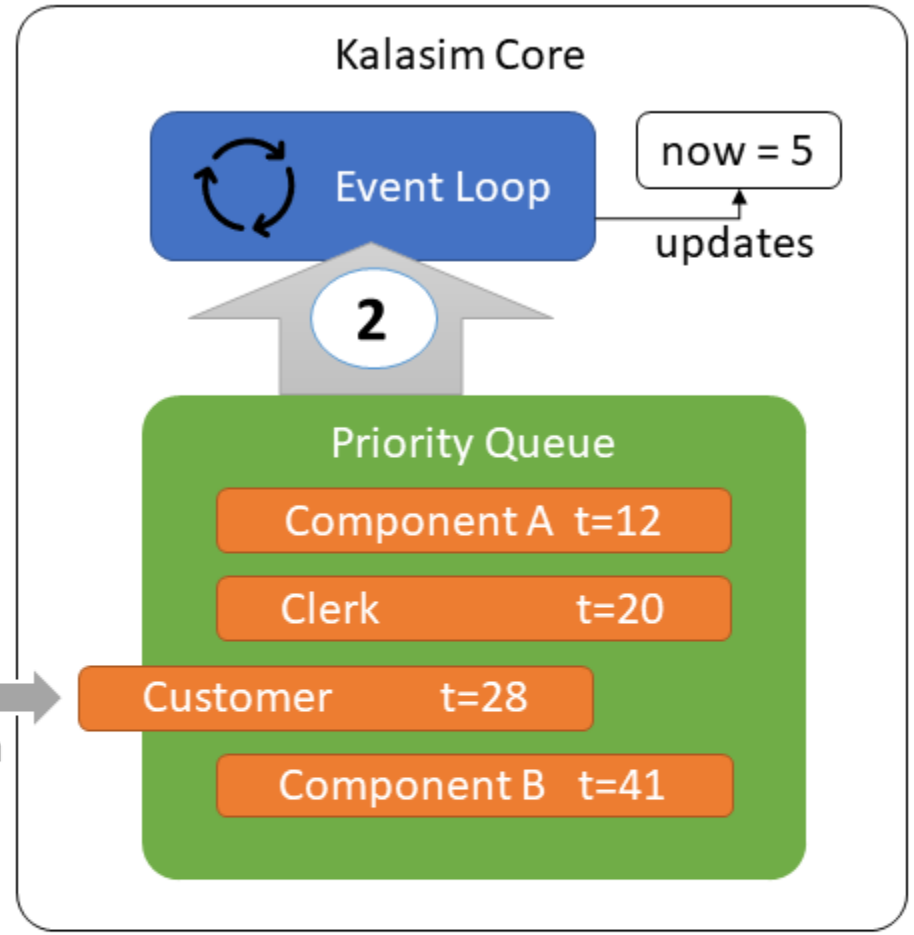
Used Interaction Methods

- `request()` – Ask (and wait) for resource
- `hold()` – Suspend execution
- `wait()` – Wait for specific state

```
////Cars.kts
import org.kalasim.*

class Driver : Resource()
class TrafficLight : State<String>("red")

class Car : Component() {

    val trafficLight = get<TrafficLight>()
    val driver = get<Driver>()

    override fun process() = sequence {
        request(driver) {
            hold(1.0, description = "driving")

            wait(trafficLight, "green")
        }
    }
}

createSimulation(enableConsoleLogger = true) {
    dependency { TrafficLight() }
    dependency { Driver() }

    Car()
}.run(5.0)
```

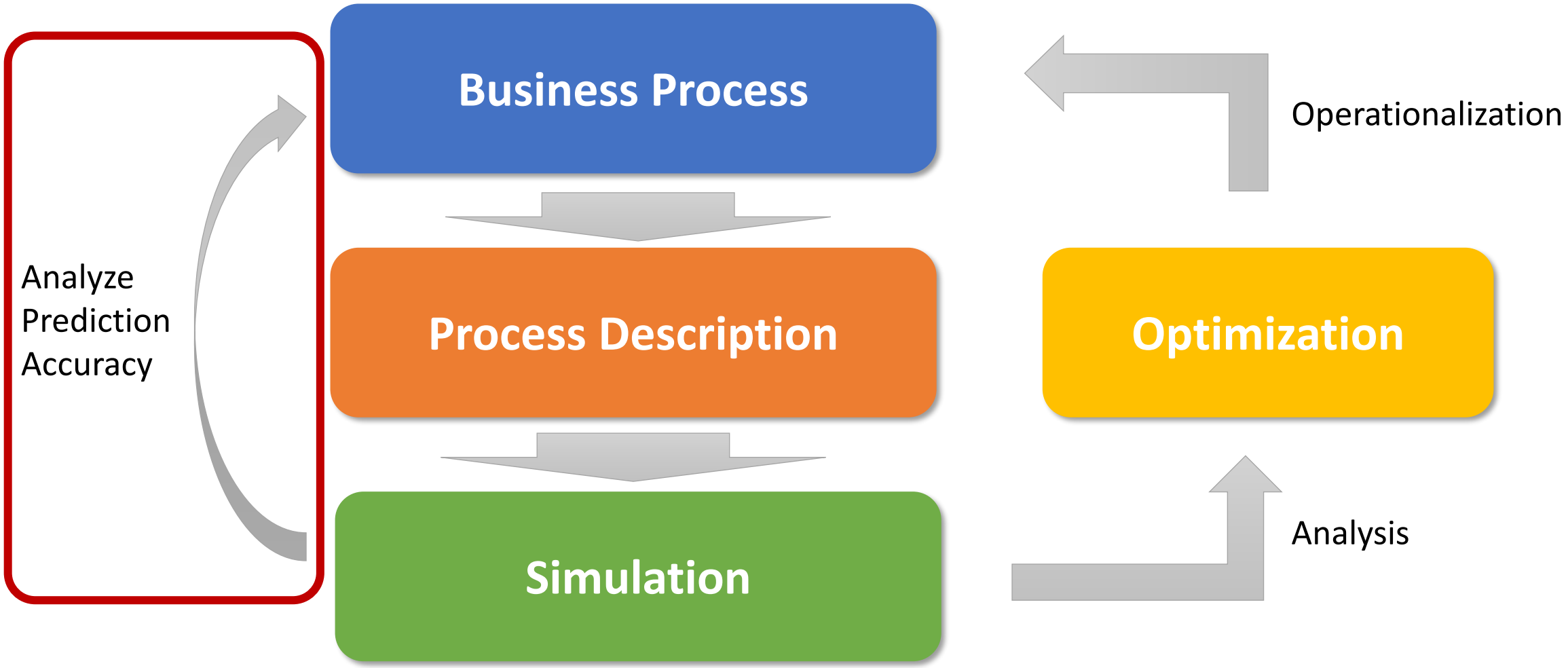
# EVENT LOOP ARCHITECTURE

```
class Customer(val clerk: Resource) : Component() {

    override fun process() :Sequence<Component> = sequence {
        // do shopping
        hold(ticks = 23.0)

        // wait for an empty counter
        request(clerk)

        // billing process
        hold(ticks = 2.0, priority = HIGH)
    }
}
```




1. Stall customer's `process` execution and insert it into event queue by time (and optional priority to resolve ambiguities)
2. Poll, while queue is not empty and resume (or start) `process`
3. Event loop will continue execution here. Components are terminated once `process` end has been reached

# SIMULATION ANALYSIS



# STUDY THE SIMULATION


Understand simulation dynamics, and play what-if together with process stake holders

# PROCESS ANALYTICS TOOLBOX INCLUDED

`kalasim` offers various means to analyze data created by a simulation

- The **Event Log** tracks events in a simulation
- **Monitors** track state and statistics of the basic elements within a simulation, and may be used for domain-specific entities as well
- **Lifecycle Records** summarize a component's states history
- **Visualization** to inspect complex spatio-temporal patterns

https://www.kalasim.org/analysis/

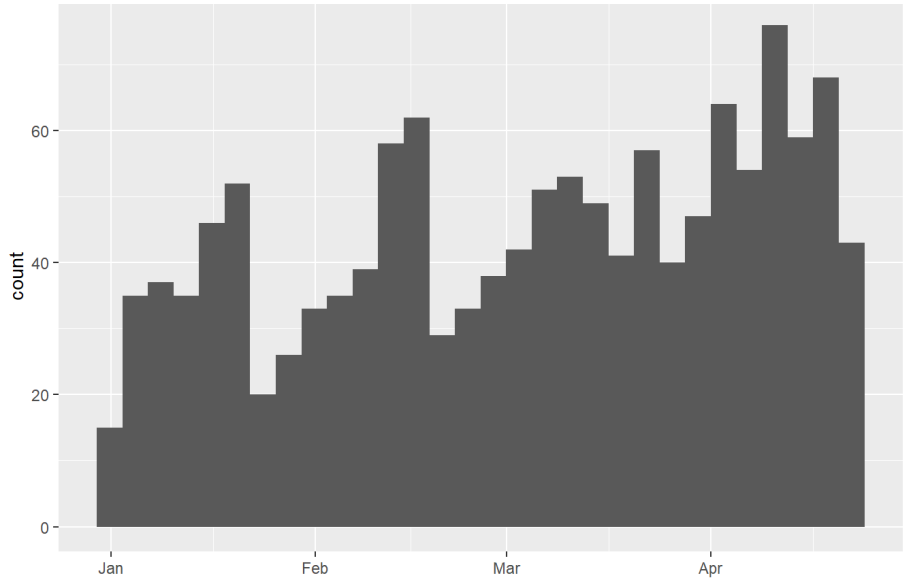
# EXPORT SIMULATION STATS FOR ANALYSIS

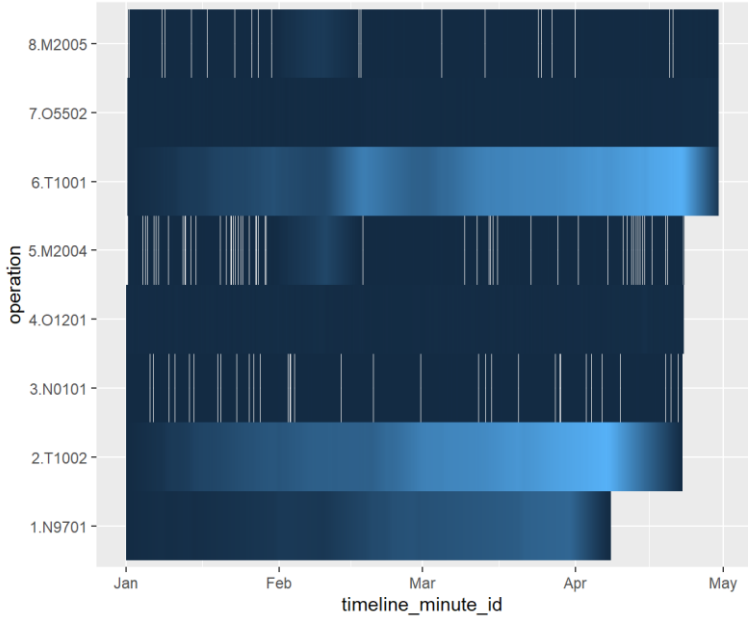
```
listOf<Whatever>()
    .asDataFrame()
    .writeCsv("simulation-data.csv")
```

- Export process metrics and analyze them (R, kotlin-kernel, python, Excel)
- Until recently via `asDataFrame` https://holgerbrandl.github.io/krangl/data_model/
- Now https://kotlin.github.io/dataframe/createdataframe.html#todataframe

# CAN WE MODEL AN FACTORIES WITH `kalasim`?

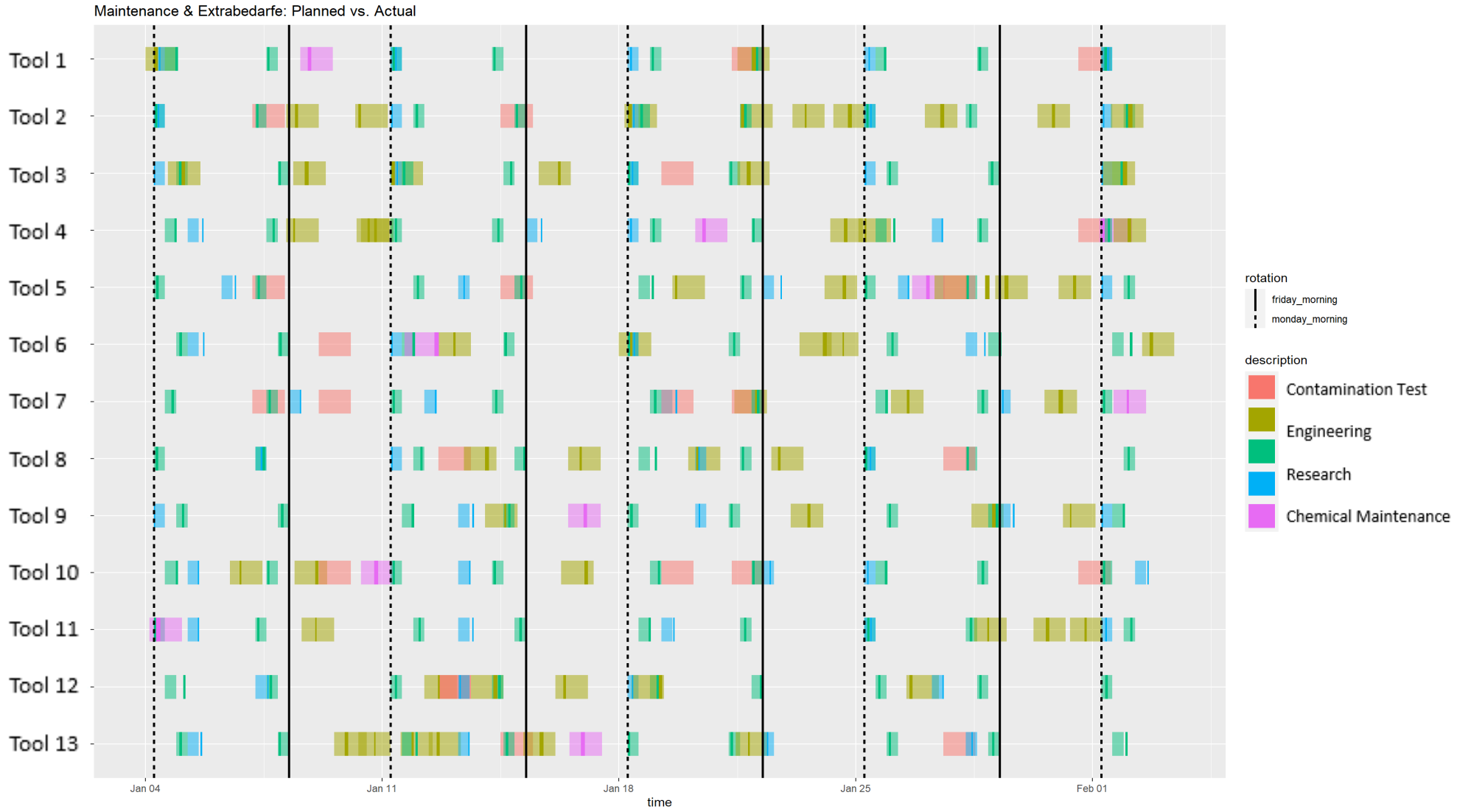
- Yes, we can!
- Can we study its dynamics? Yes, we can.
- Can we use these models to suggest/test process improvements. Yes, we can.



EXAMPLE

# MAINTENANCE IN SIMULATED PRODUCTION AREA




Note: Tool & task names are redacted

# ANALYZE IN PLACE WITH KOTLIN-JUPYTER

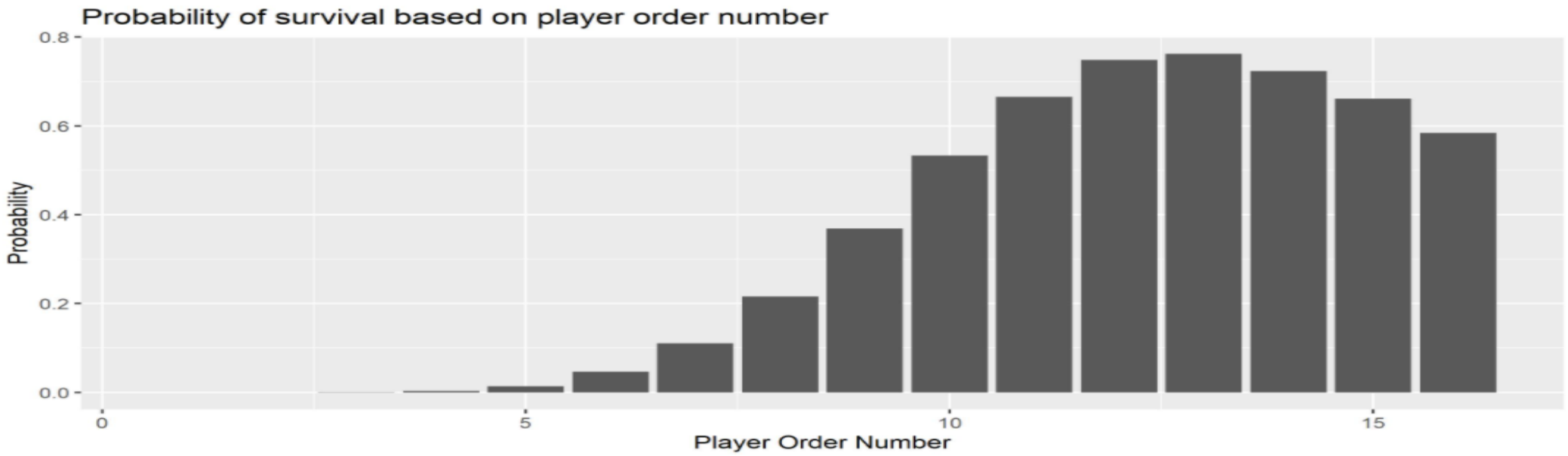
jupyter nbviewer

JUPYTER FAQ

In [92]:

```
val survivalProbByNo = (1..manyGames.first().numPlayers).map { playerNo ->
    playerNo to manyGames.count { it.playerSurvived(playerNo) }.toDouble() / manyGames.size
}

survivalProbByNo.plot(x = { it.first }, y = { it.second }).geomCol().labs(
    title = "Probability of survival based on player order number",
    x = "Player Order Number",
    y = "Probability"
)
```

Out[92]:

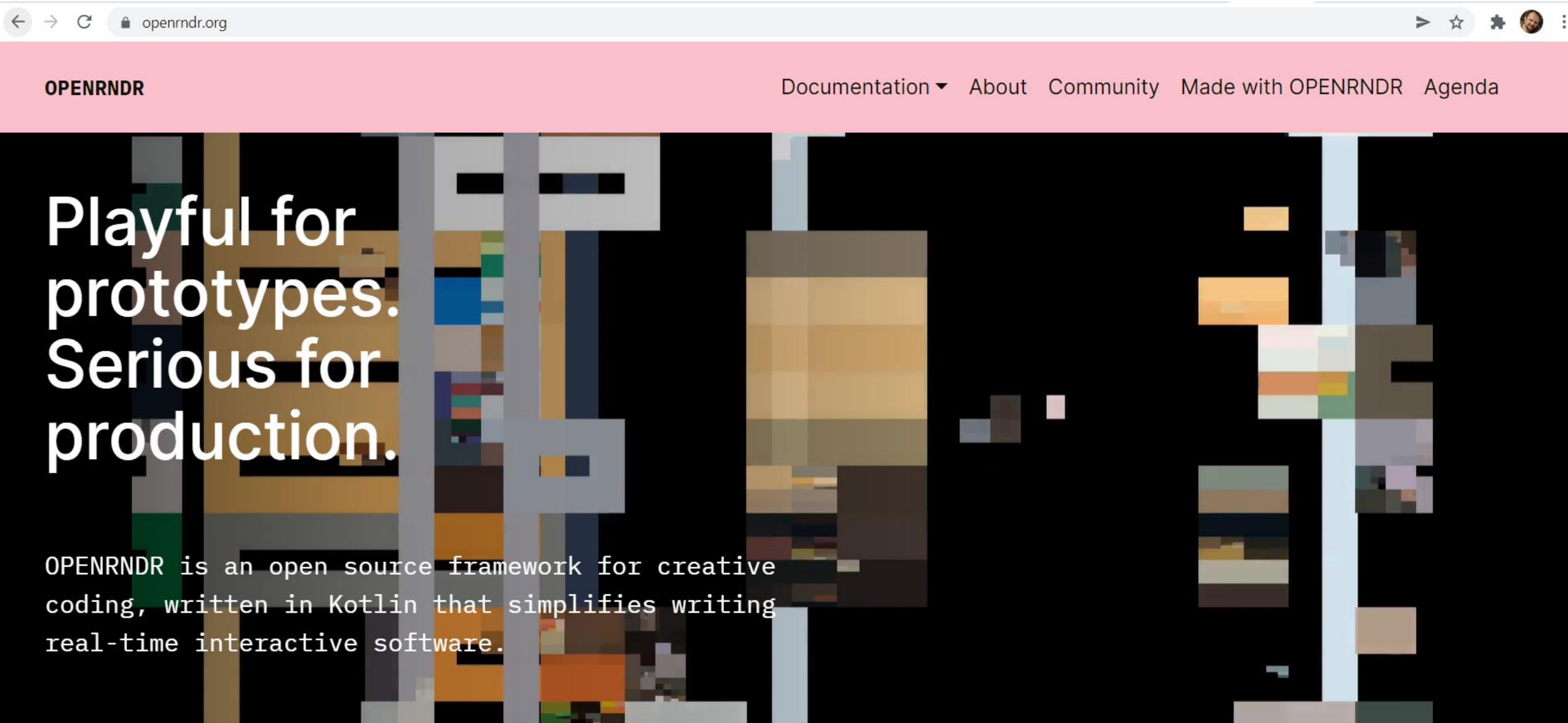



So indeed there seems a strategy to maximize your odds of survival in the game. Simply pick **No13**, and you may live more likely compared to any other starting number.

https://www.kalasim.org/examples/gas_station/

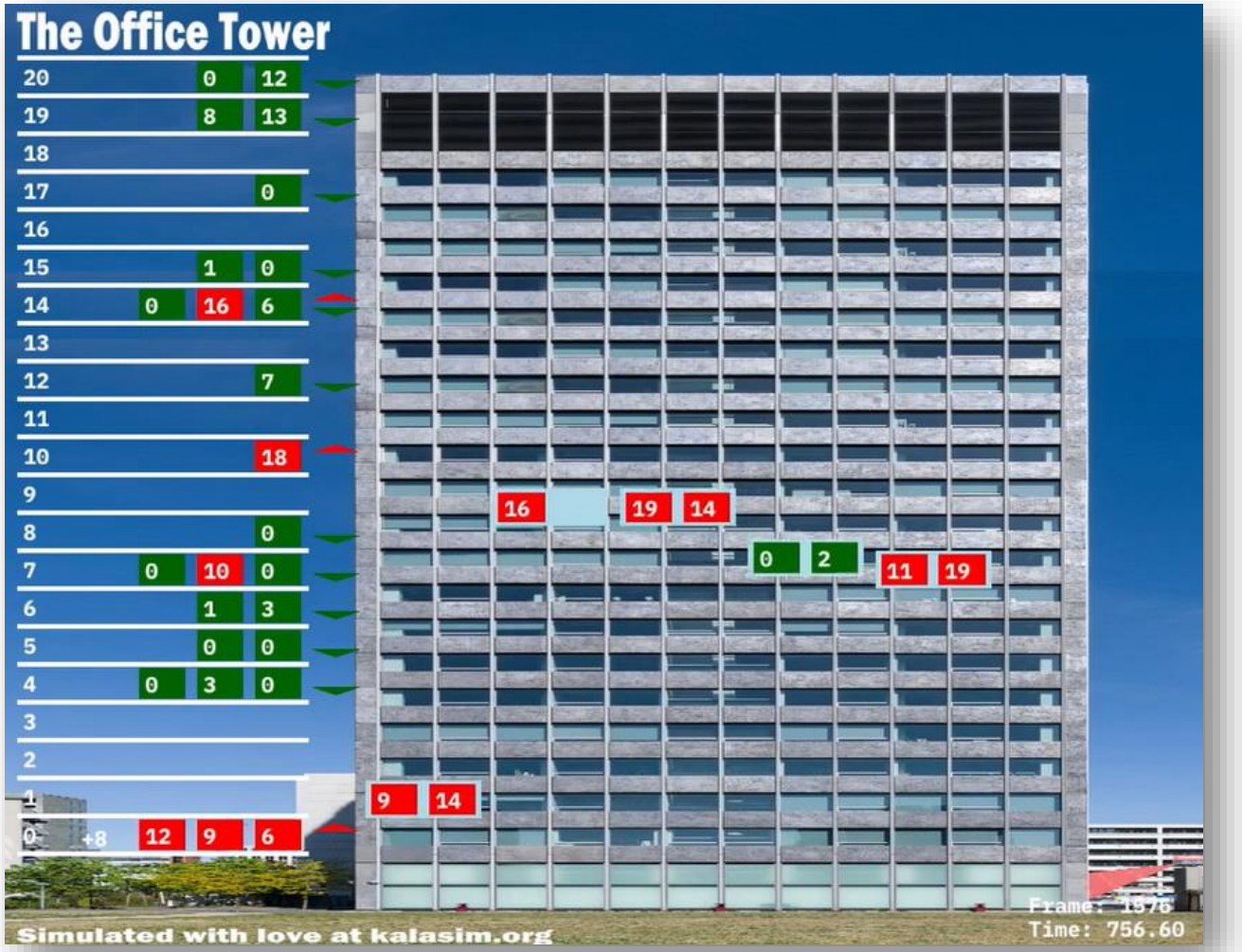
https://www.youtube.com/watch?v=r2igemRWk7s

# PROCESS ANIMATION → OPENRNDR



# OPENRNDR API WALKTHROUGH

- Drawing primitives
- Animating primitives
- Superimposing a grid system
- Kalasim support API
- Example Office Tower



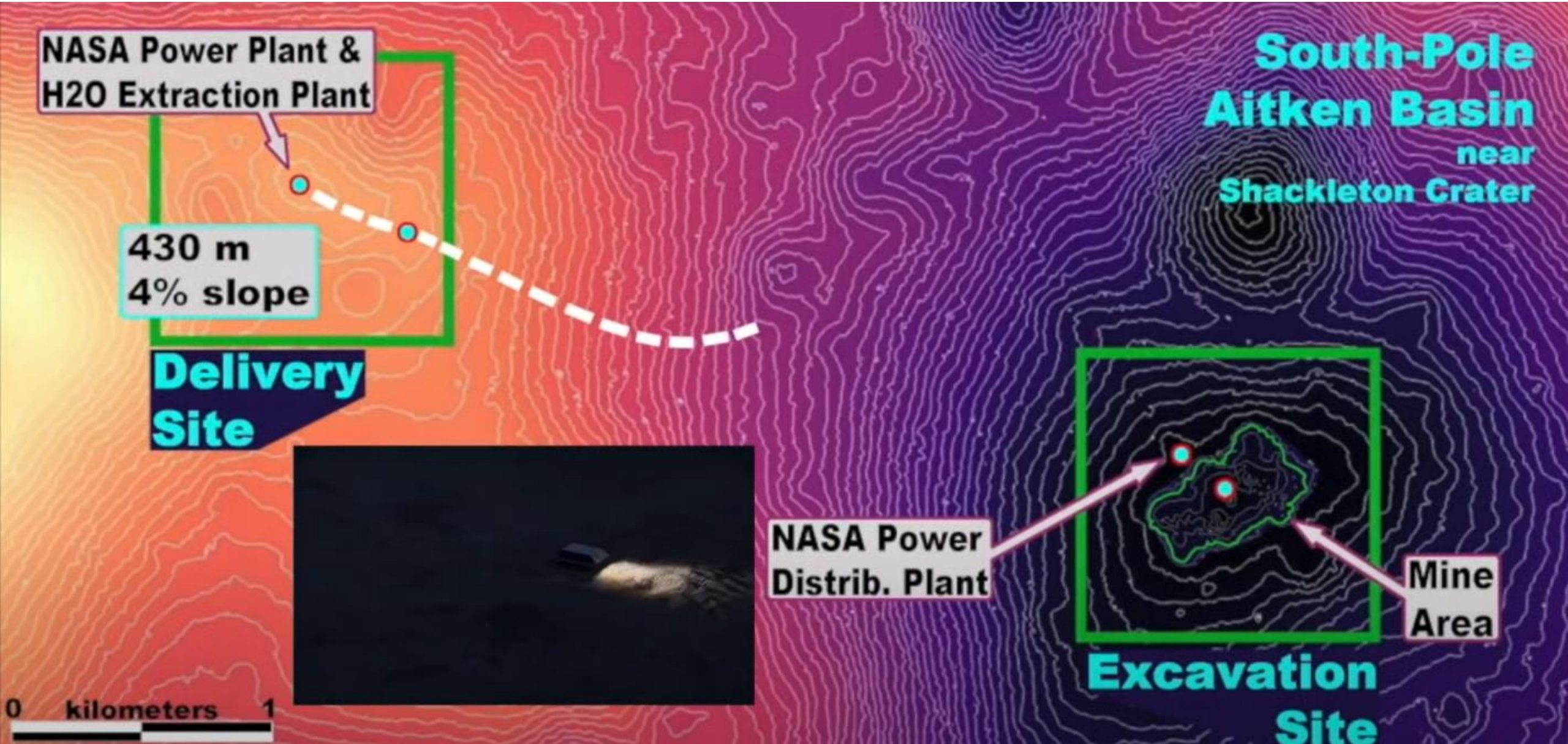

https://www.kalasim.org/examples/office_tower/

https://www.youtube.com/watch?v=Hk2copteiaE

# NASA'S BREAK THE ICE LUNAR CHALLENGE

https://www.youtube.com/watch?v=Hk2copteiaE

# REMOTE MINING SITES



# SIMULATION MODEL OBJECTIVES

ESPA has ordered their process specialists to work out a simulation model of the mining process. With the simulation, the number of mining robots needed to supply the base with enough water must be determined. Also, water production rates shall be estimated.

ESPA simulation engineers have to solve two very typical tasks in industrial engineering

1. Capacity Planning (number of mining robots needed)
2. Forecast of Production KPIs (tons of water/day)

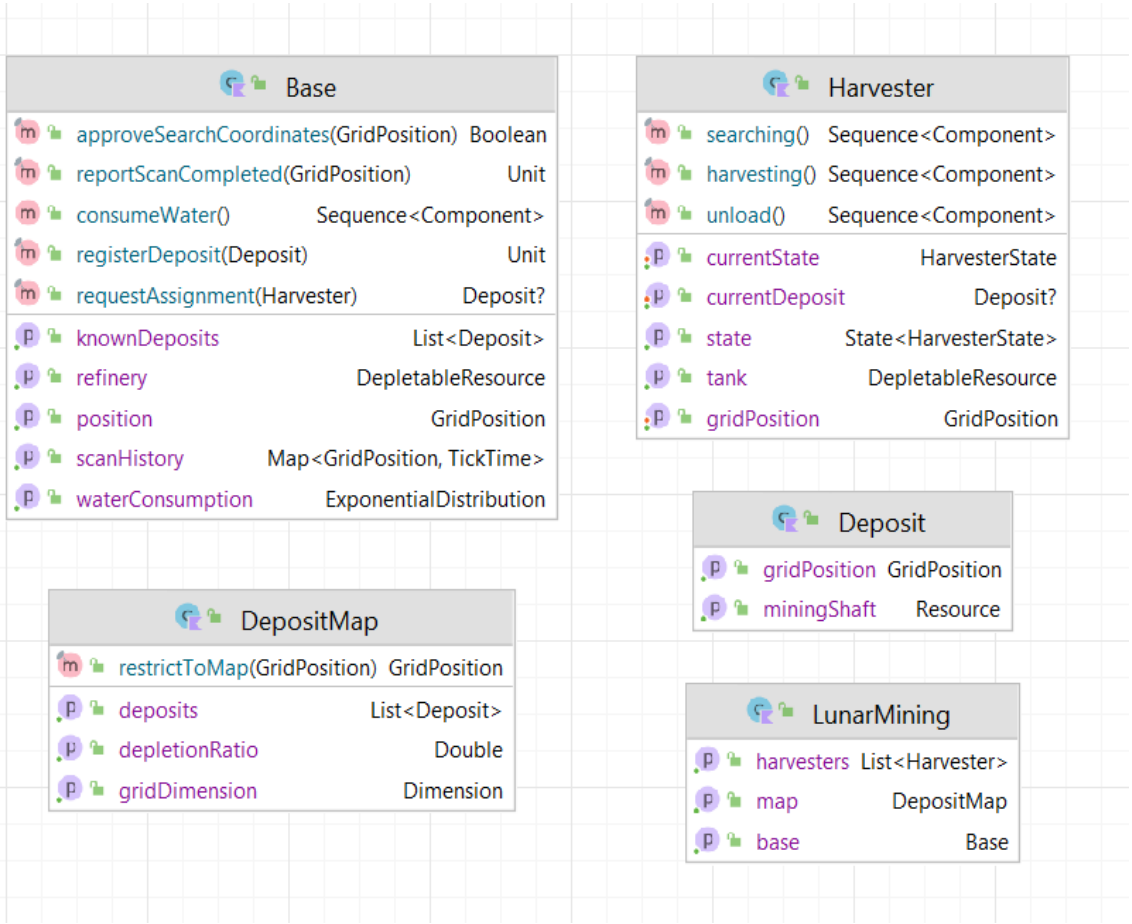
# PROCESS DESCRIPTION: LUNAR MINING

- There is a complex interplay of rate-limited processes (transport, search, mining), limited resources and the harsh realities of deep space. The latter is abstracted away in the model as it does not contribute to model performance
- While the specific locations of ice-deposits are unknown, their average distribution and size on the moon had been determined already using a satellite equipped with the onboard radar, ultraviolet detectors as well as a neutron spectrometer
- Small harvester robots are being deployed from a central depot to scan the lunar surface for water deposits
- When finding a depot they deplete it
- They have a limited storage capacity (100kg), so they will need to shuttle the cargo to the base
- The base will consume water constantly (exponentially distributed with a mean of 5 kg/h)
- The base has an initial deposit of 100kg water (which was shipped to the moon very expensively with rockets from earth)
- Idle harvesters will consult the base for nearby deposits discovered by other units

https://www.kalasim.org/animation/lunar_mining/

# LUNAR MINING PROCESS

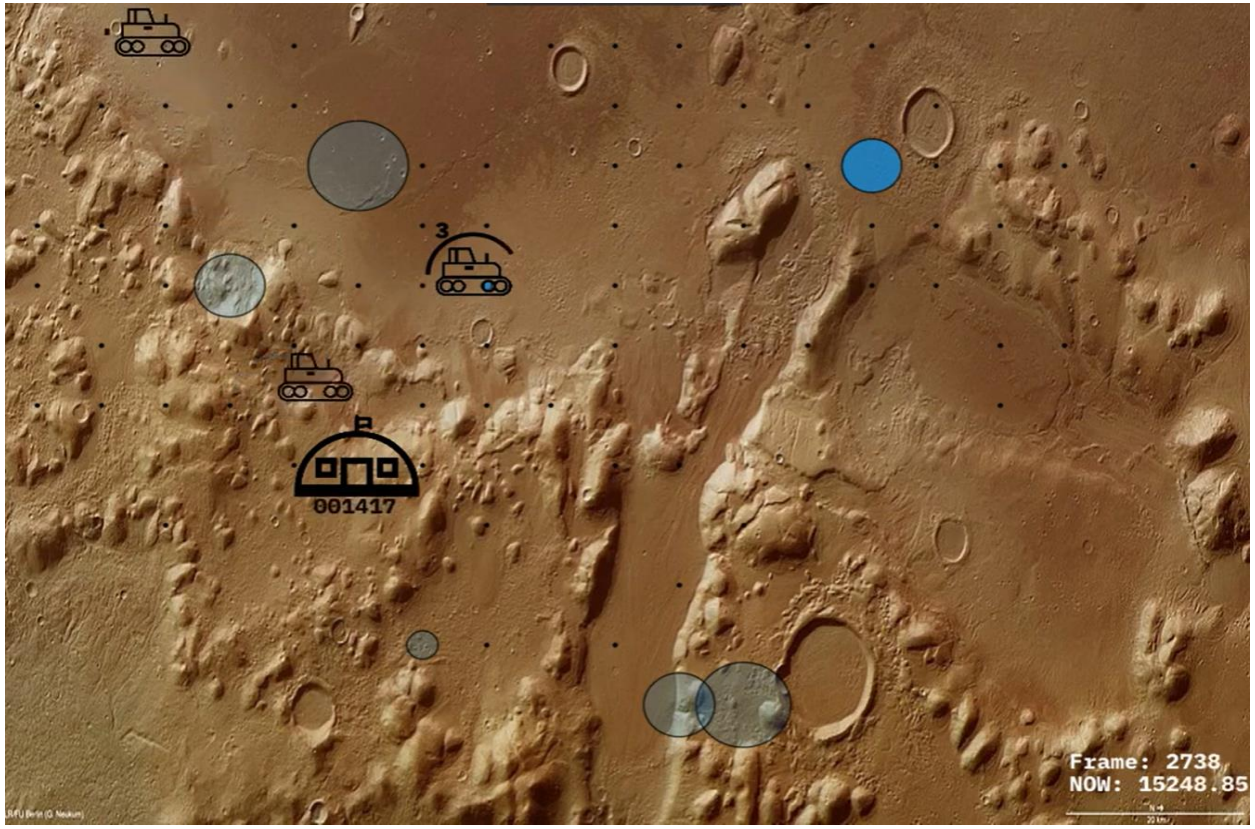
Mining robots scan the surface of the moon for depletable water ice deposits.
It is a complex business process, and the lives of thirsty astronauts are at stake.



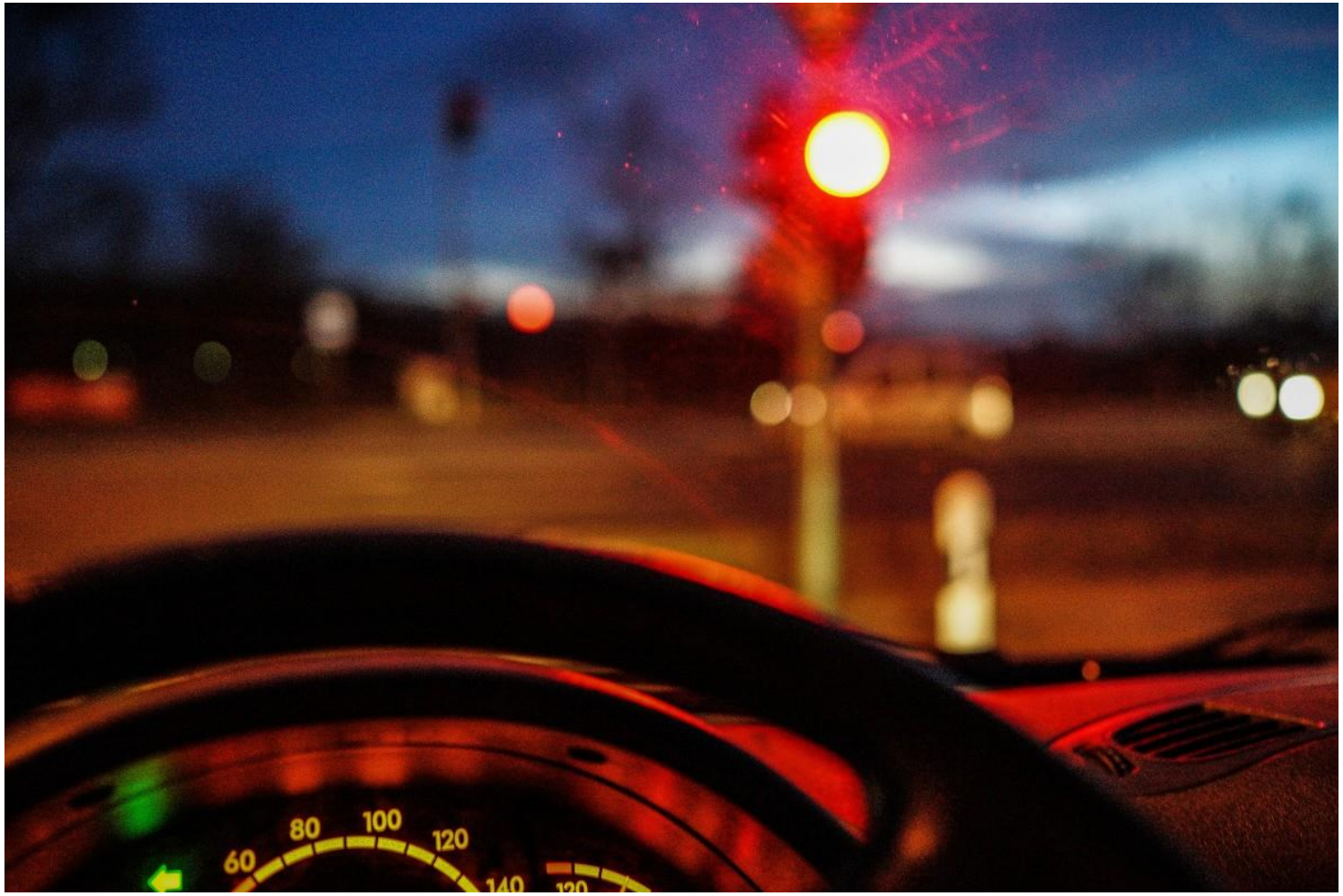

```
fun unload() = sequence {
    moveTo(base.position)

    val unloadingUnitsPerHours = 20  // speed of unloading

    // unloading time correlates with load status
    currentState = UNLOADING
    hold((tank.level / unloadingUnitsPerHours).roundToInt().hours,
         "Unloading ${tank.level} water units")

    // put the water into the refinery of the base
    put(get<Base>().refinery, tank.level)

     // empty the tank
    take(tank, tank.level)

    activate(process = Harvester::harvesting)
}
```

# LUNAR MINING IN A BOX

- Jupyter Notebook → https://nbviewer.org/github/holgerbrandl/kalasim/blob/master/docs/userguide/docs/animation/lunar_mining.ipynb
- Lunar Mining Animated →



# EXERCISE 1 - TRAFFIC




https://datalore.jetbrains.com/view/notebook/hgUbUBCOv0peUUinqL86A3

# Kotlin / kotlin-jupyter Public



master 10 branches 51 tags Go to file Add file Code

ileasile Update Kotlin version ... af410ab 17 days ago 887 commits

| | | |
|---|---|---|
| .idea/dictionaries | Add integration for londogard-nlp-toolkit library | 10 months ago |
| .run | Add detection of the current Jupyter Client to the API. Switch build ... | last month |
| additional-licenses | Add JLaTeXMath library to lib-ext | 11 months ago |
| api-examples/getting-started | Update Kotlin to 1.6.0, KSP to 1.0.1 and Gradle to 7.3 | 3 months ago |
| binder | Update versions | 4 months ago |
| build-plugin | Add detection of the current Jupyter Client to the API. Switch build ... | last month |
| distrib-util | Add python requirements installing to Gradle build | 2 years ago |
| distrib | Try to fix conda test | 4 months ago |
| docs | Update Kotlin version | 17 days ago |
| gradle | Update Kotlin version | 17 days ago |

## About

Kotlin kernel for Jupyter/IPython

Readme

Apache-2.0 License

663 stars

35 watching

89 forks

## Releases

51 tags

## Packages

No packages published

## Distplot

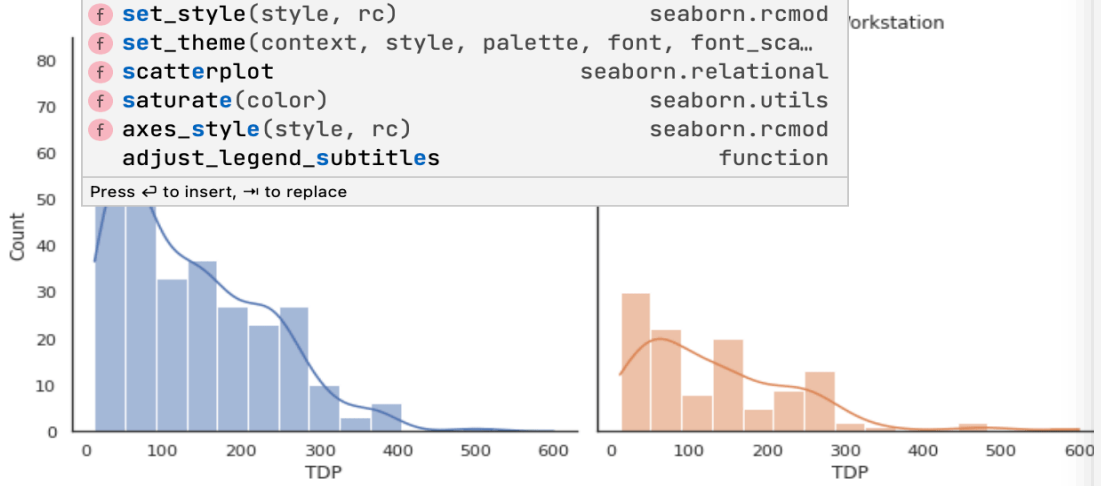
Another interesting topic is GPU power consumption and heating.

While server GPUs are more hot and powerful on average than consumer and workstation ones, it's interesting that there's a lot of low-power workstation GPUs. These ones are intended not for 3D graphics or parallel computing, but for managing display panels and video walls.

[1] 3.9s

```
import pandas as pd
import seaborn as sns

sns.se
df = p
sns.di                                                    rue)
```

```
f set(args, kwargs)                          seaborn.rcmod
f set_color_codes(palette)                 seaborn.palettes
f set_context(context, font_scale, r…       seaborn.rcm…
f set_hls_values(color, h, l, s)             seaborn.utils
f set_palette(palette, n_colors, desat, colo…   seab…
f set_style(style, rc)                       seaborn.rcmod
f set_theme(context, style, palette, font, font_sca…
f scatterplot                             seaborn.relational
f saturate(color)                            seaborn.utils
f axes_style(style, rc)                      seaborn.rcmod
  adjust_legend_subtitles                        function
Press ⏎ to insert, ⇥ to replace
```



# KOTLIN-KERNEL - STILL WORK-IN-PROGRESS

- Works great, unless you hit one of its walls

```
[1]
import kotlin.time.Duration.Companion.seconds

[5]
3.seconds

Back-end (JVM) Internal error: Couldn't inline method call: seconds
Method: null
File being compiled: (1,3) in Line_10.jupyter-kts
The root cause java.lang.IllegalStateException was thrown at: org.jetbrains.kotlin.codegen.inline.Sou

org.jetbrains.kotlinx.jupyter.repl.impl.JupyterCompilerImpl.compileSync(JupyterCompilerImpl.kt:171)
org.jetbrains.kotlinx.jupyter.repl.impl.InternalEvaluatorImpl.eval(InternalEvaluatorImpl.kt:95)
org.jetbrains.kotlinx.jupyter.repl.impl.CellExecutorImpl$execute$1$result$1.invoke(CellExecutorImpl.
org.jetbrains.kotlinx.jupyter.repl.impl.CellExecutorImpl$execute$1$result$1.invoke(CellExecutorImpl.
org.jetbrains.kotlinx.jupyter.ReplForJupyterImpl.withHost(repl.kt:602)
org.jetbrains.kotlinx.jupyter.repl.impl.CellExecutorImpl.execute(CellExecutorImpl.kt:63)
org.jetbrains.kotlinx.jupyter.repl.CellExecutor$DefaultImpls.execute$default(CellExecutor.kt:13)
```

Filters | is:issue is:open label:"REPL involved" | Labels 15 | Milestones 1 | New issue

Clear current search query, filters, and sorts

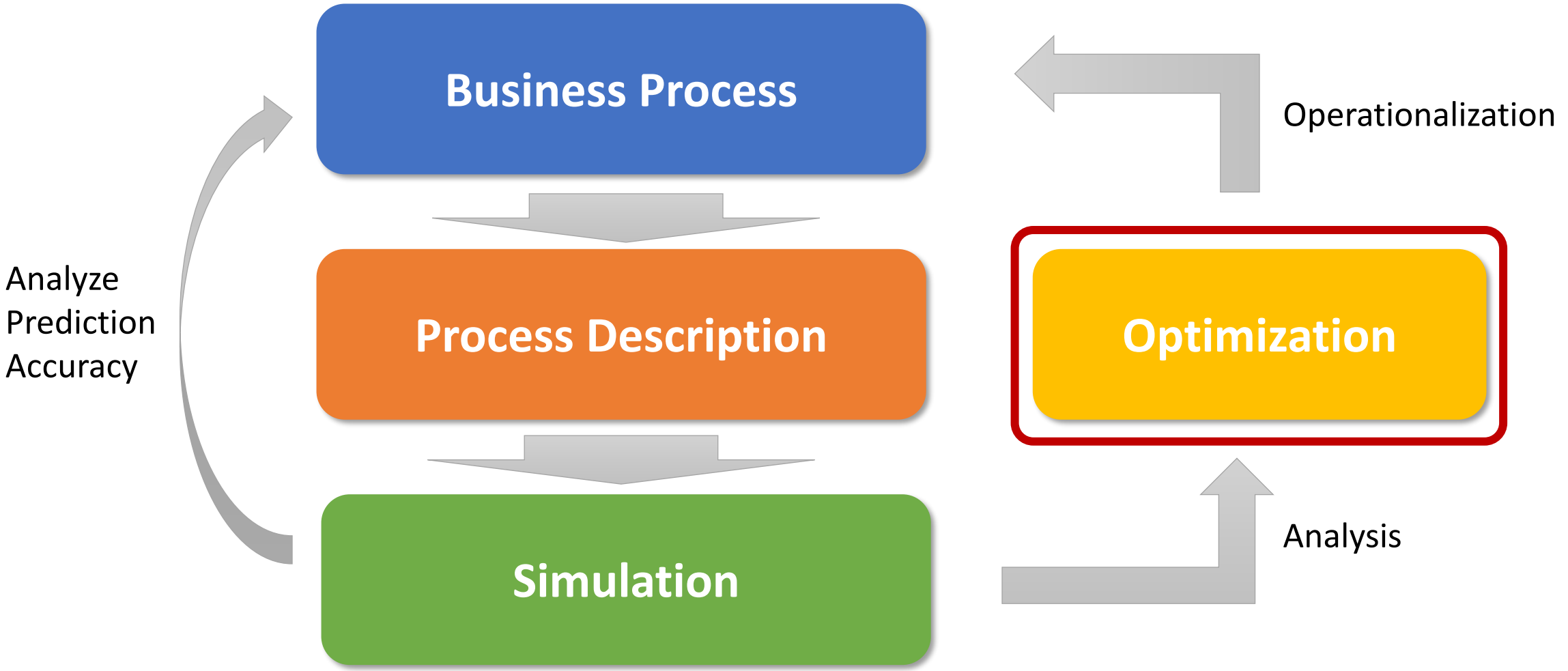
7 Open | 5 Closed | Author | Label | Projects | Milestones | Assignee | Sort

- method 'void <init>()' not found [bug] [REPL involved] — 1
  #355 opened 29 days ago by holgerbrandl
- Can not wildcard import enums [bug] [REPL involved] — 3
  #353 opened on Jan 15 by holgerbrandl
- Back-end (JVM) Internal error: Failed to generate expression: KtNameReferenceExpression [bug] [duplicate] [REPL involved] — 1
  #345 opened on Nov 15, 2021 by holgerbrandl
- Autocompletion omits type parameters even when they're mandatory [bug] [REPL involved]
  #178 opened on Mar 26, 2021 by strangepleasures
- Failed to generate expression: KtNameReferenceExpression when assigning property in object [bug] [REPL involved] — 3
  #126 opened on Feb 11, 2021 by sergeygrigorev
- Documentation popup [feature] [REPL involved] — 1
  #73 opened on May 27, 2020 by altavir
- The kernel dies when using variables with different registers. [bug] [REPL involved] — 1
  #44 opened on Feb 4, 2020 by devcrocod

# USE OF SIMULATION FOR PROCESS OPTIMIZATION

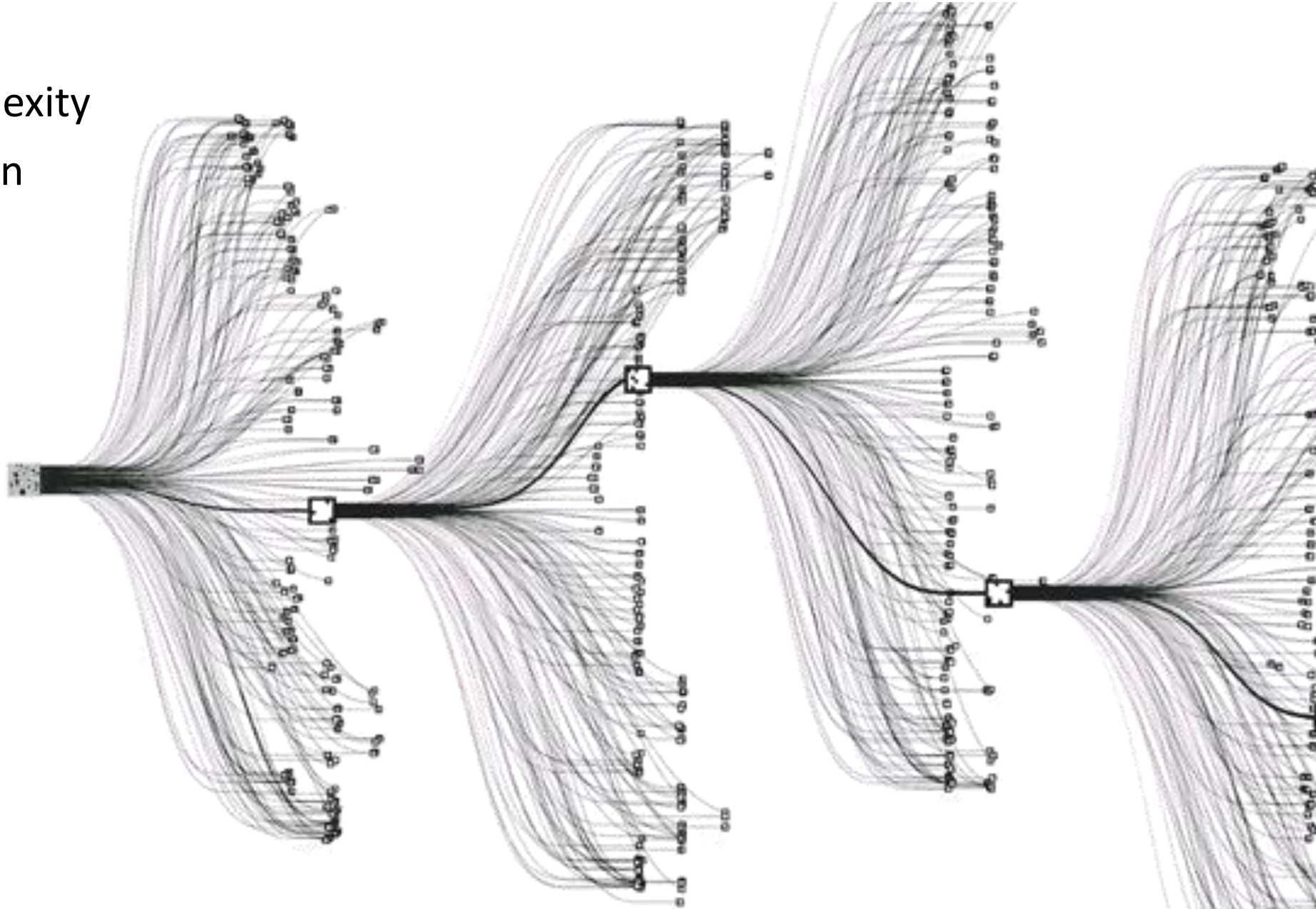
## RISK FREE PROCESS CHANGE ASSESSMENT

# PROCESS OPTIMIZATION LIFECYCLE



# WHY IS MANUFACTURING EXECUTION PLANNING HARD?

- Search space complexity
- Problem formulation
- Factory dynamics
- Data requirements
- NP-hard



# BENEFITS OF ADVANCED PRODUCTION PLANNING

Some benefits of optimized production scheduling include

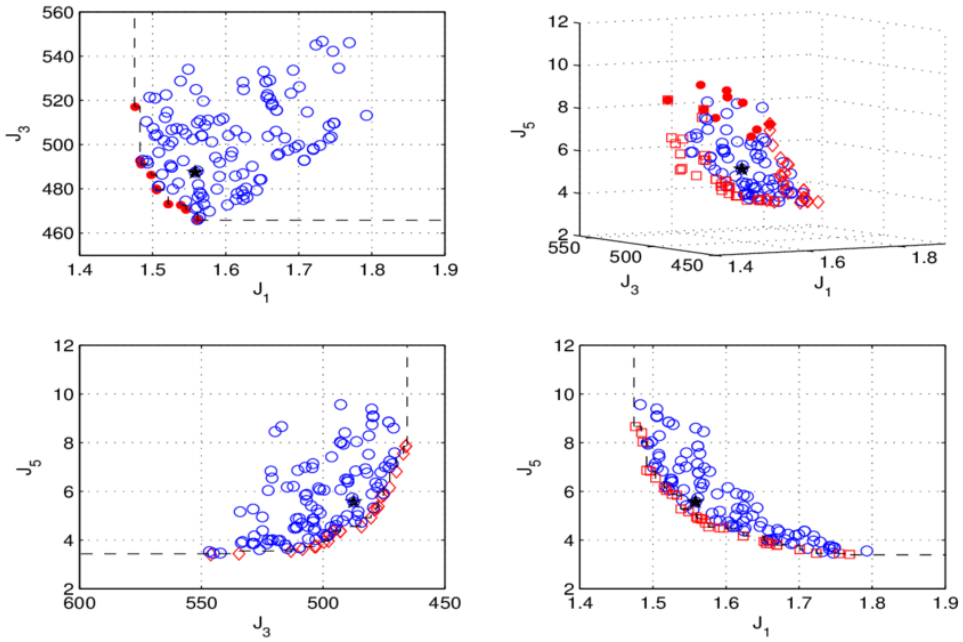
- Increased production efficiency
- Setup optimization
- Cycle-time reduction
- Process change-over reduction
- WIP balancing & Inventory reduction
- Personnel load levelling & Shift planning
- Accurate delivery date prediction

# OPTIMIZATION OBJECTIVES IN MANUFACTURING

- Setup avoidance
- Minimize doping change transitions
- Avoid running dry
- Maximize throughput
- Minimize transport

→ Multi-objective optimization

Always: Go for the low hanging fruits first!

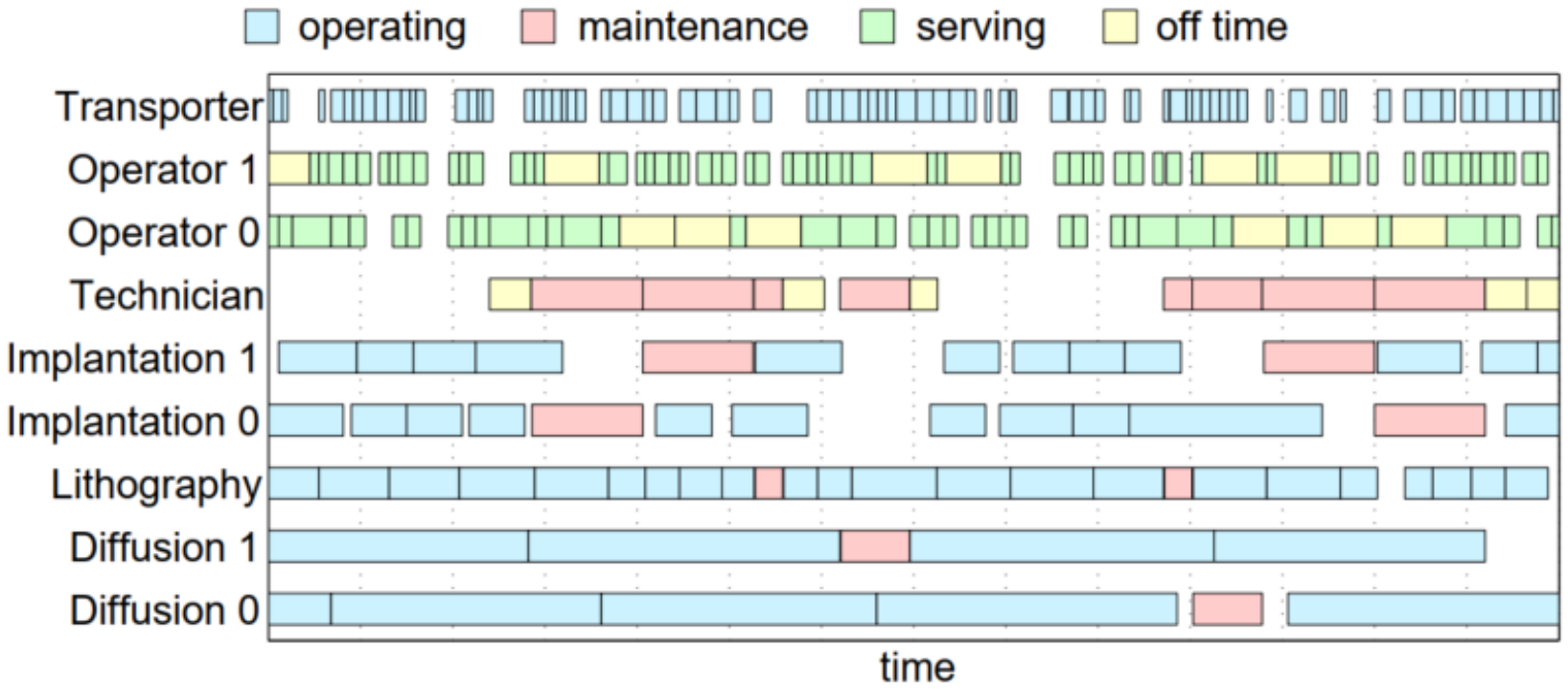


# 4M RESOURCE ALIGNMENT

4M – Machine, Method, Material & Human

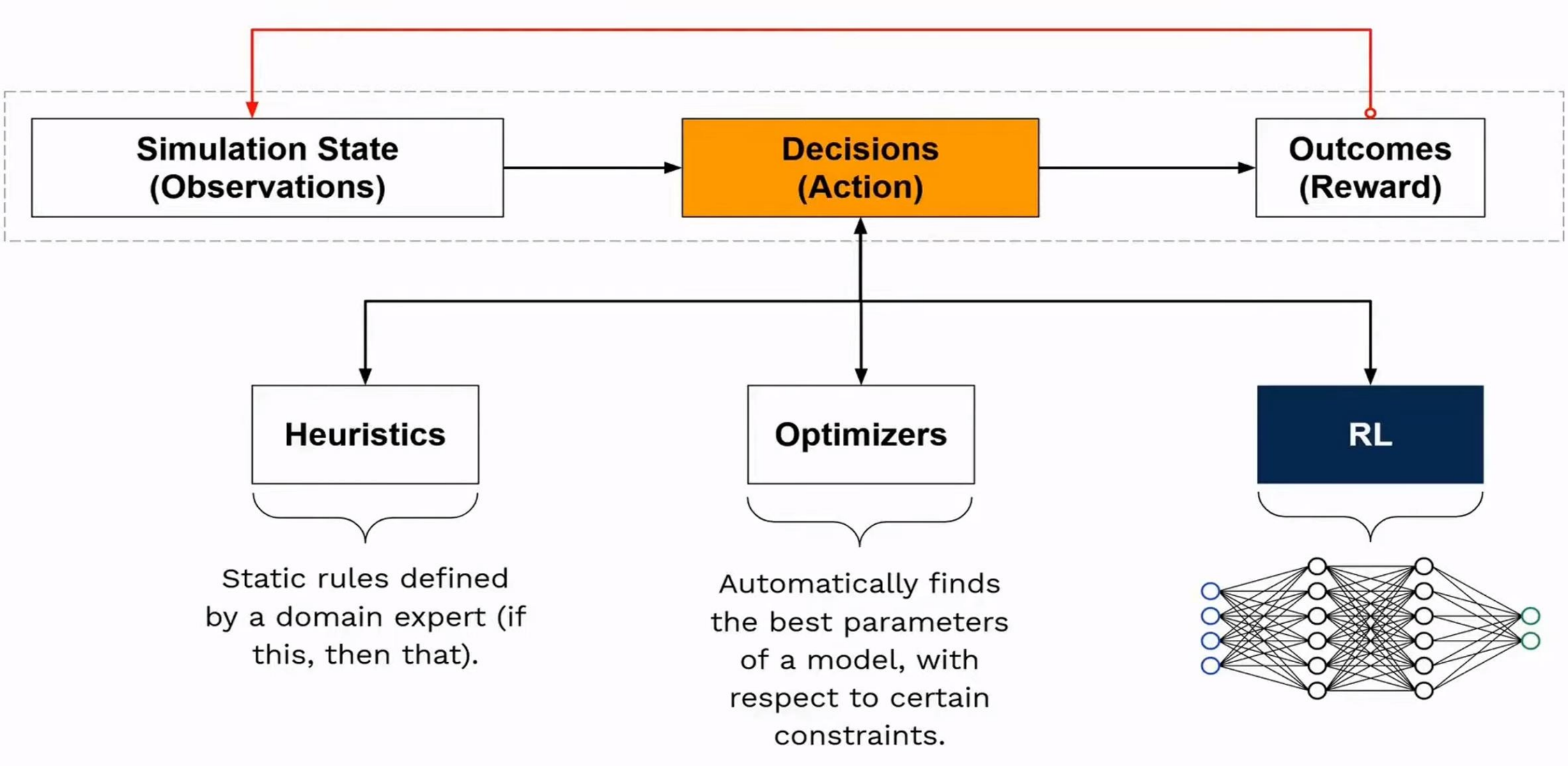
Process optimization is a lot about resource alignment

Make sure that material, method, human and machine are available at the same time

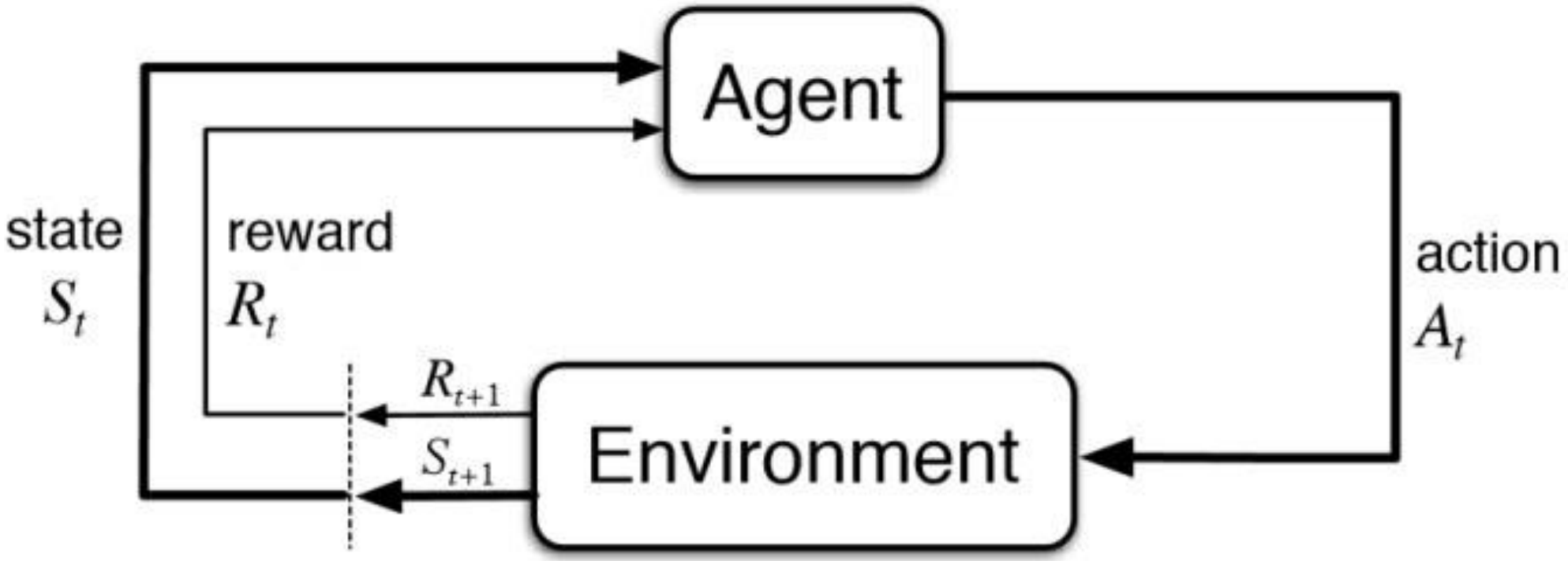


Source: http://pathmind.com

# METHOD DEVELOPMENT FOR BETTER PLANNING

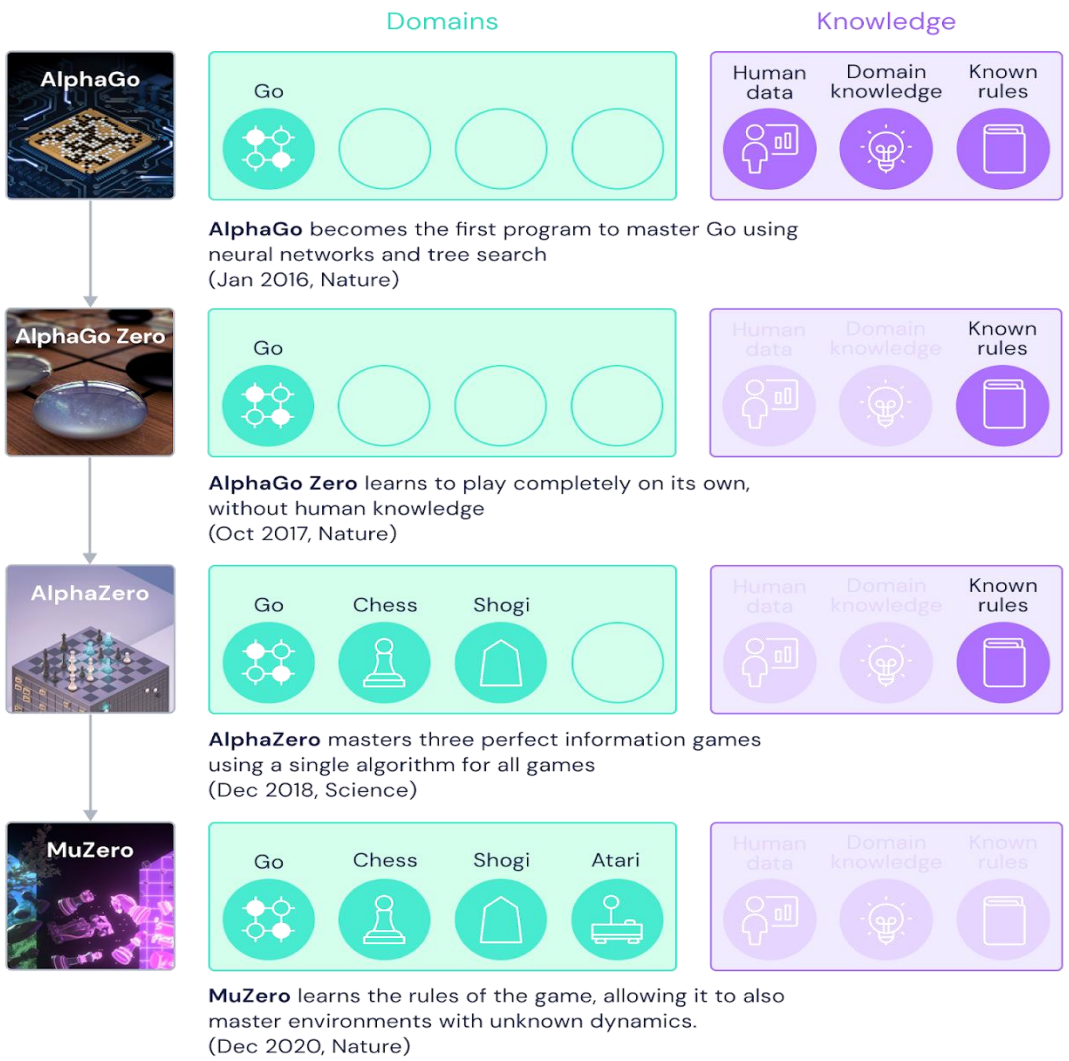


# REINFORCEMENT LEARNING




Source

# AI-BASED PROCESS CONTROL

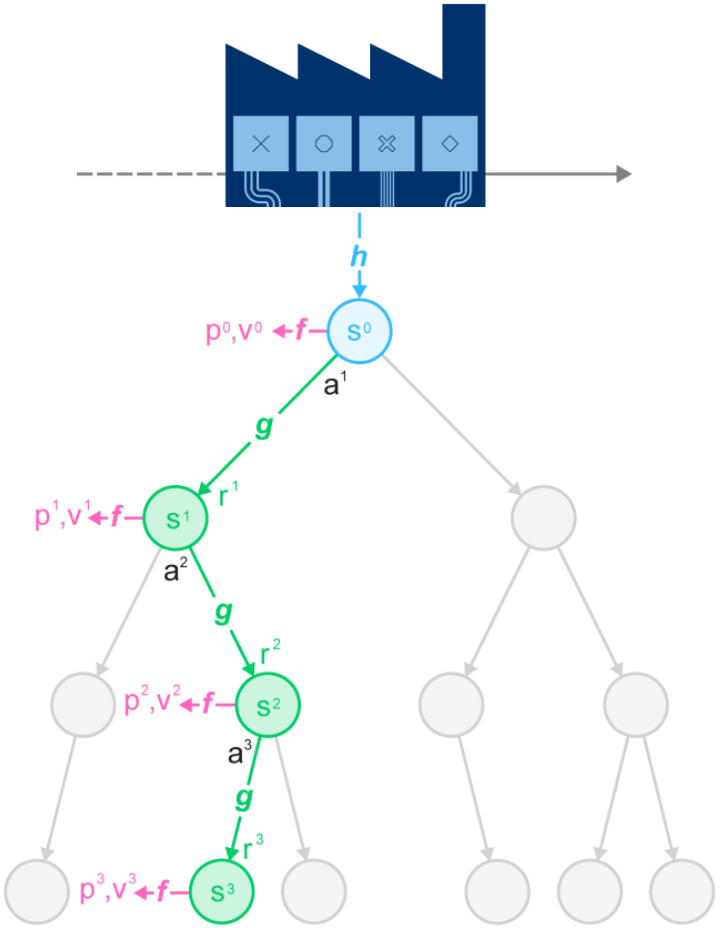


# "MU MANUFACTURING" ☺

- Automate planning and execution to run production at scale
- Find the most effective execution plan for any given business objective with AI

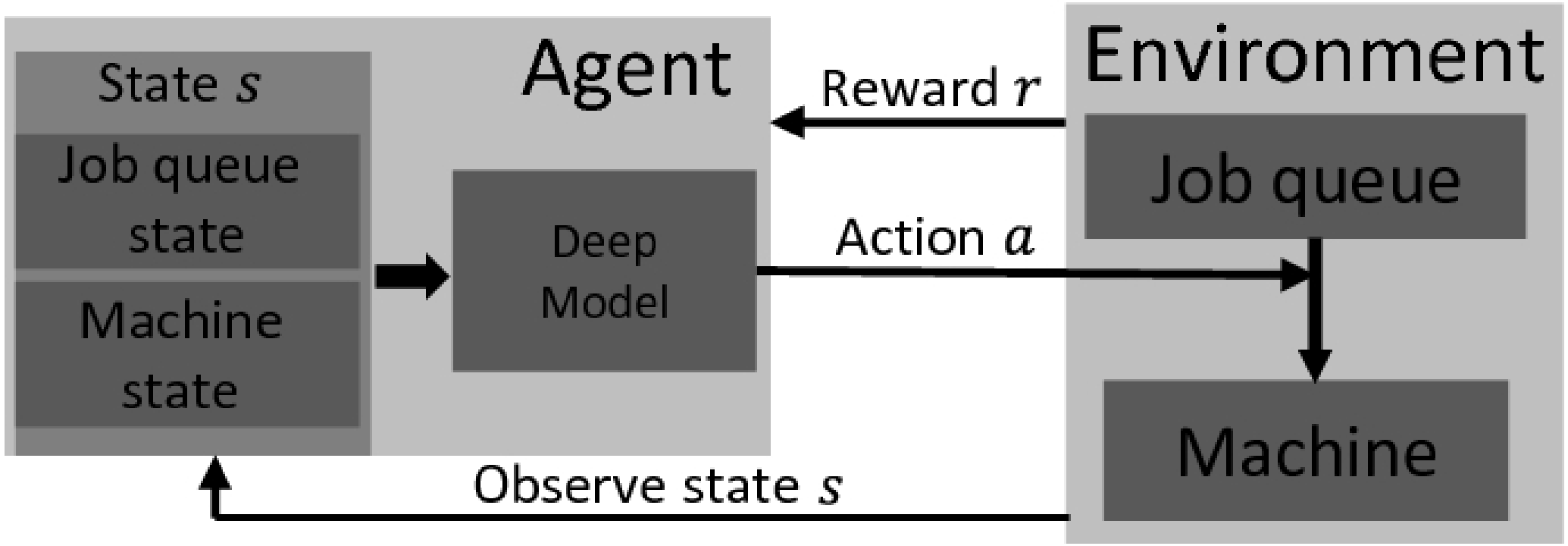
- **Value**: Due date adherence
- **Policy**: Material routes, shift plans
- **Reward**: OEE

- But: Many unknowns



# REINFORCEMENT LEARNING IN PRODUCTION SCHEDULING

- Research in Progress → Operationizable > 2025+



Source: NIPS 2015

# REQUIREMENTS FOR MACHINE LEARNING OPERATIONALIZATION (ML OPS)

- How to bring models into production in production?
- How to reflect dynamics (product changes)
- How to track model quality?
- How to update models?

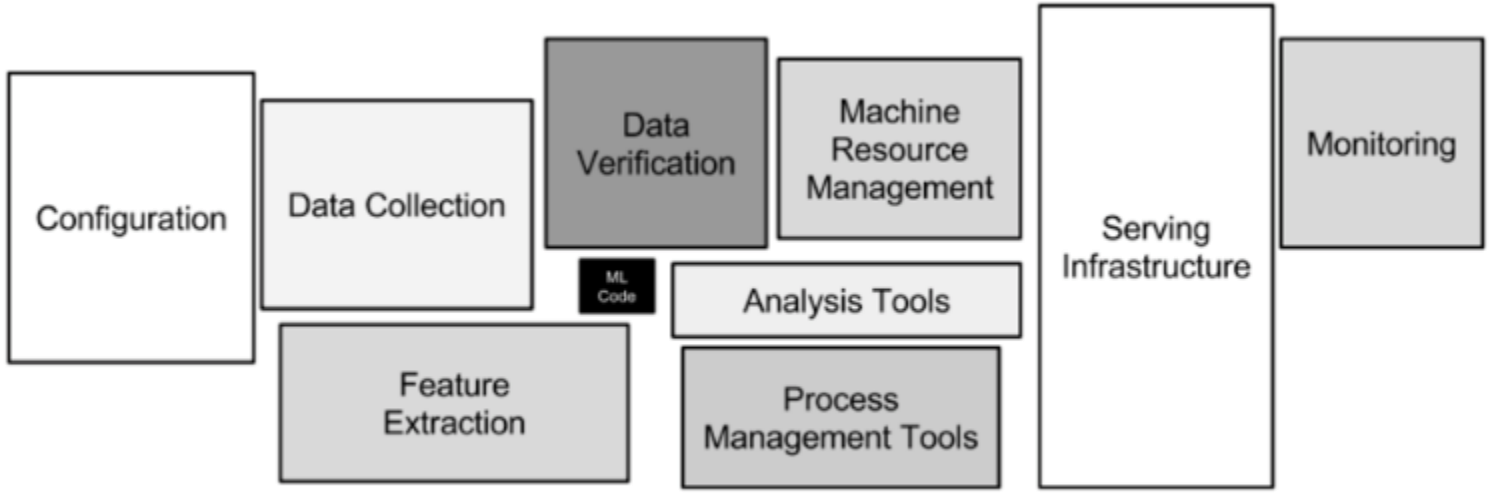
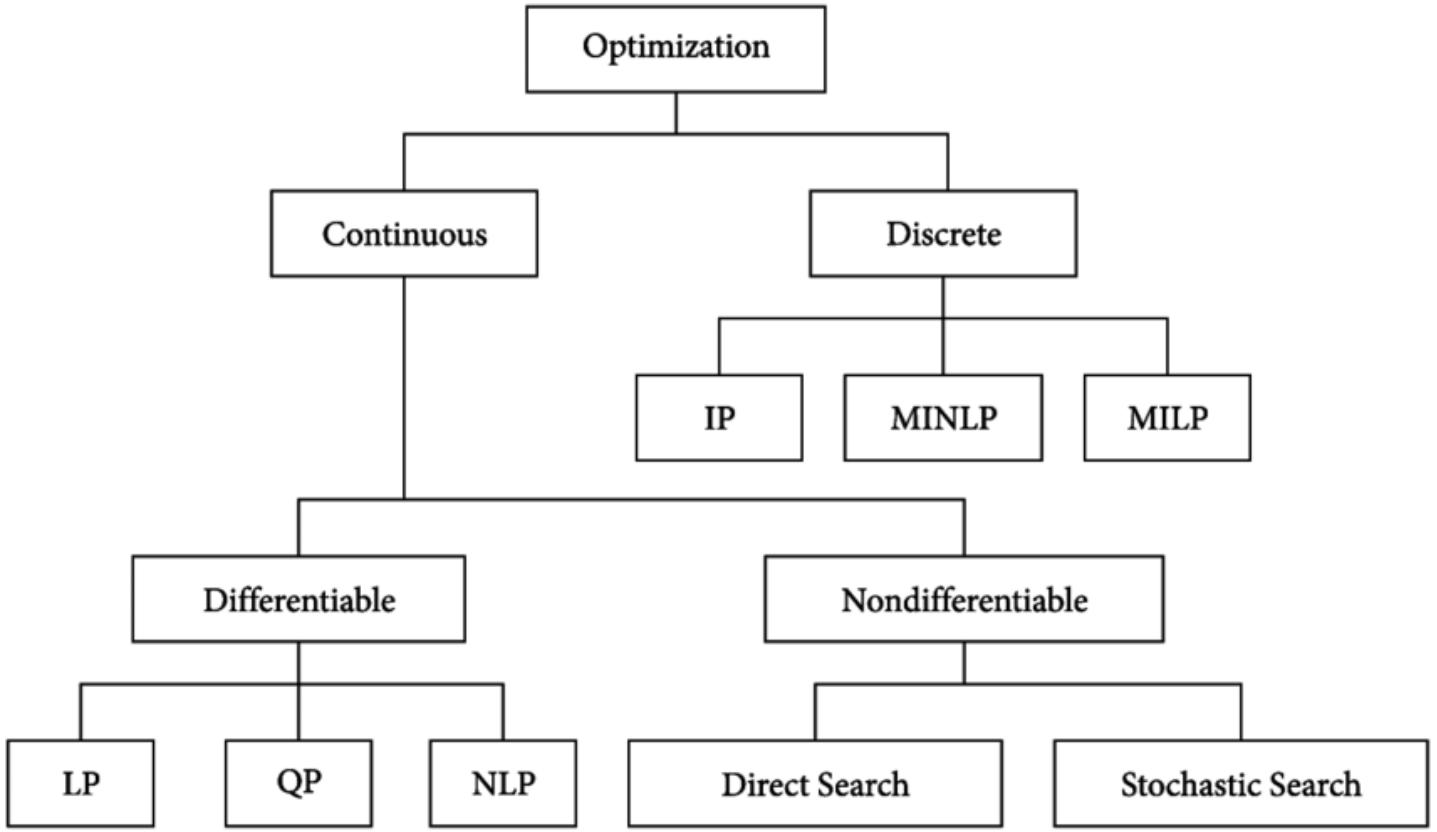



Figure 1. Hidden Technical Debt in Machine Learning Systems

# MATHEMATICAL OPTIMIZATION

- Mixed-integer linear or nonlinear programming (MILP/MINLP)

$$\begin{aligned} \text{maximize} \quad & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{subject to} \quad & A\mathbf{x} + \mathbf{s} = \mathbf{b}, \\ & \mathbf{s} \geq \mathbf{0}, \\ & \mathbf{x} \geq \mathbf{0}, \\ \text{and} \quad & \mathbf{x} \in \mathbb{Z}^n, \end{aligned}$$



# CONSTRAINED PLANNING OPTIMIZATION - WHAT'S THE PROBLEM?

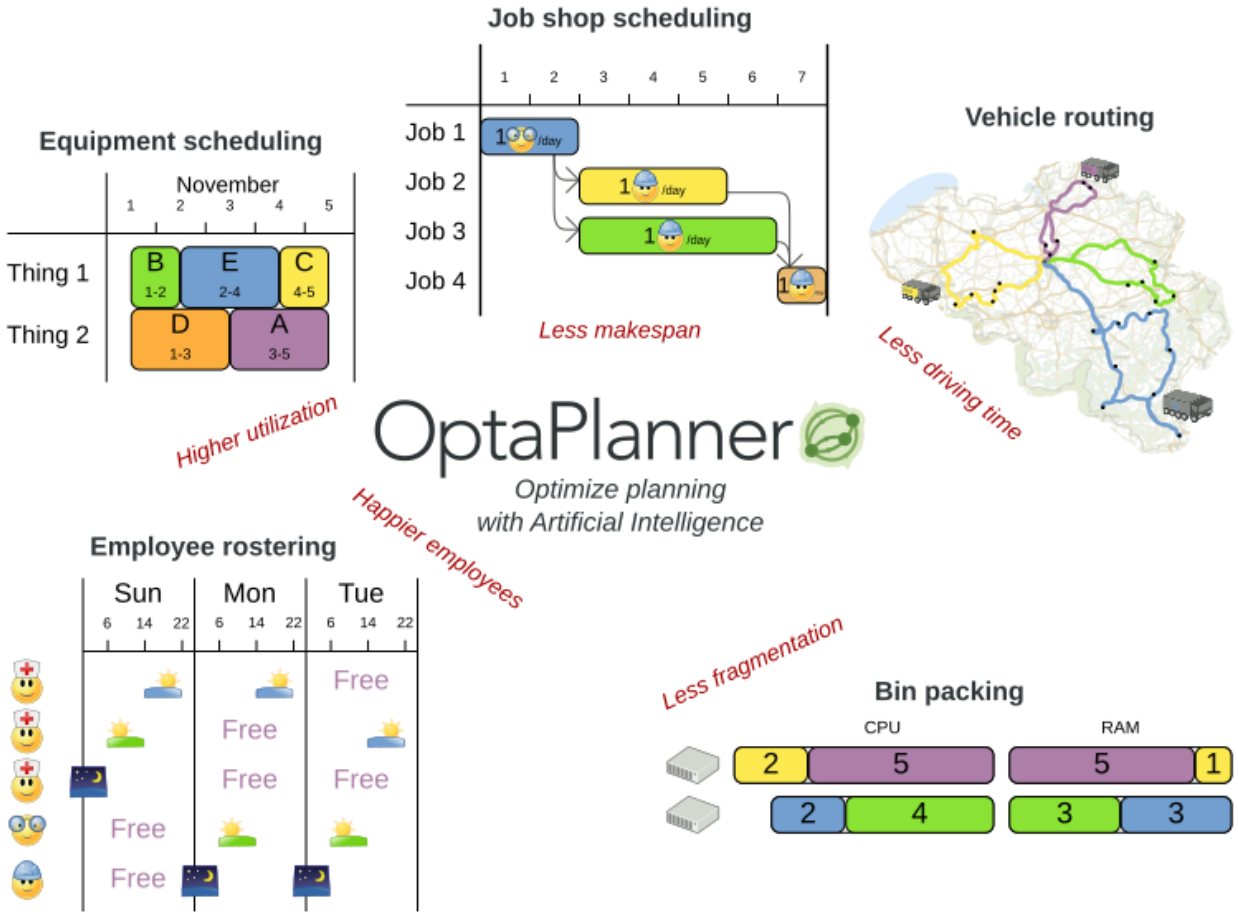
Optimize goals with limited resources under constraints


Source: https://www.optaplanner.org/

# CONSTRAINED OPTIMIZATION WITH OPTAPLANNER

- **OptaPlanner** is an AI constraint solver.
- Written in Java, fun with Kotlin

- Lightweight & embeddable
- Flexible constraints definitions
- Over-constrained planning
- Continuous planning
- Pinnable planning entities




Source: https://www.optaplanner.org/

# KALASIM APPLICATIONS

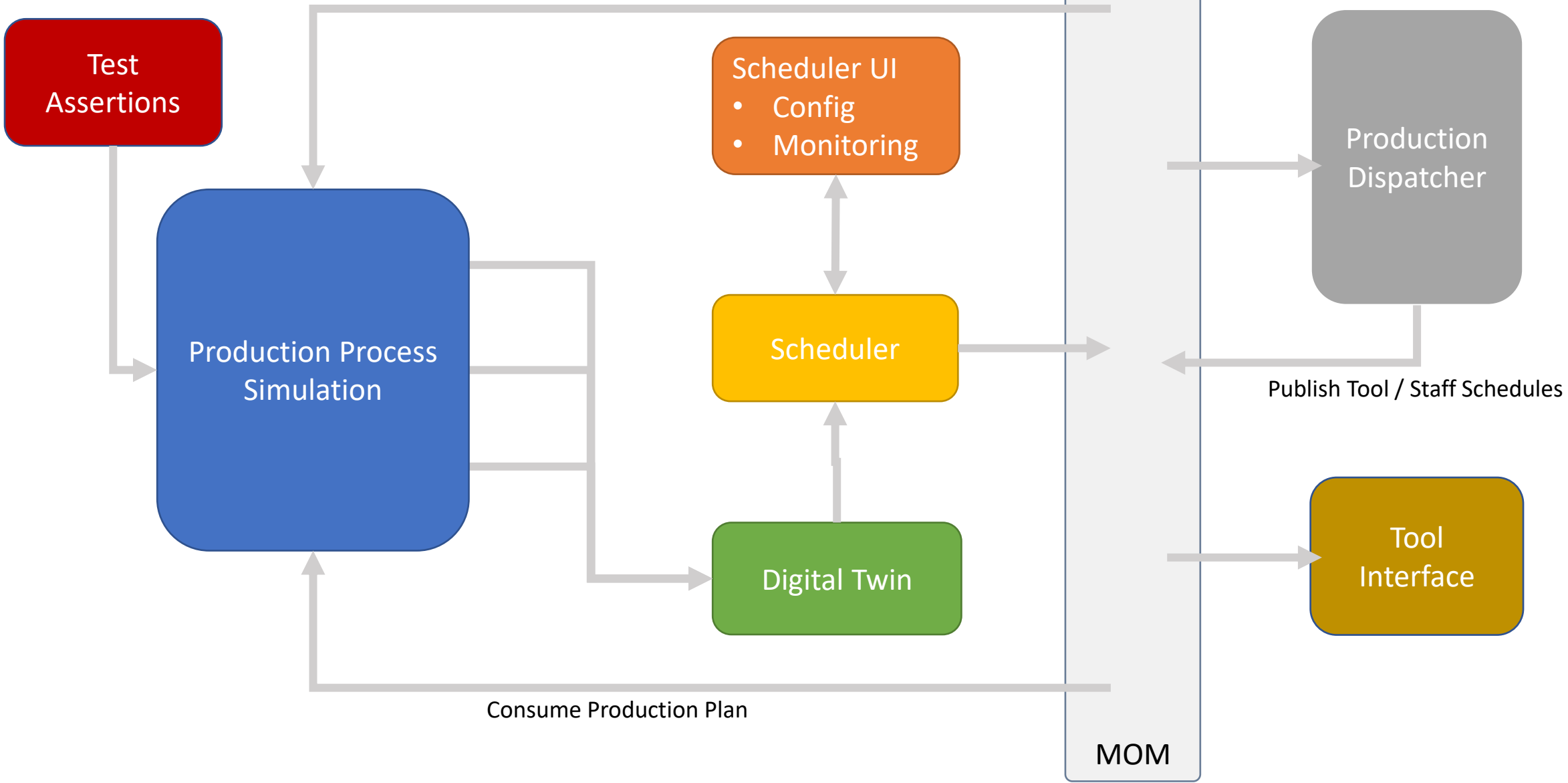
- Optimization
- AI Research
- Bottleneck Analytics
- Integration Testing
- Games & Fun
- Capacity Planning

# REAL-TIME SIMULATION

- Slow down simulation to run them in real-time (or in sync with whatever clock)

```kotlin
import org.kalasim.ClockSync
import org.kalasim.createSimulation
import kotlin.time.Duration.Companion.seconds

val timeBefore :Long = System.currentTimeMillis()

createSimulation( enableConsoleLogger: true) { this: Environment
    // enable real-time clock synchronization
    ClockSync(tickDuration = 1.seconds)

    run( duration: 10)
}

println("time passed ${System.currentTimeMillis() - timeBefore})")
```

# USE SIMULATION TO MOCK-TEST ON STEROIDS



# EXERCISE 2 - GAS STATION

- Simulate the processes at a gas station

# GAS STATION PROCESS DESCRIPTION

- Each customer requires a service agent, there is no self-service
- A service agent will do the refuel where he requires a fuel pump and a counter to do the billing
- Workers take a 10minute break once an hour
- The paper roll in the counter needs to be replaced every 100 bills which takes 15minutes

- There are 4 workers,  3 fuel pumps, 1 counter
- New customers on average (exponentially distributed) every 2 minutes
- Refueling takes 3 minutes, billing 1
- What's the approximate occupancy of the staff when there's?

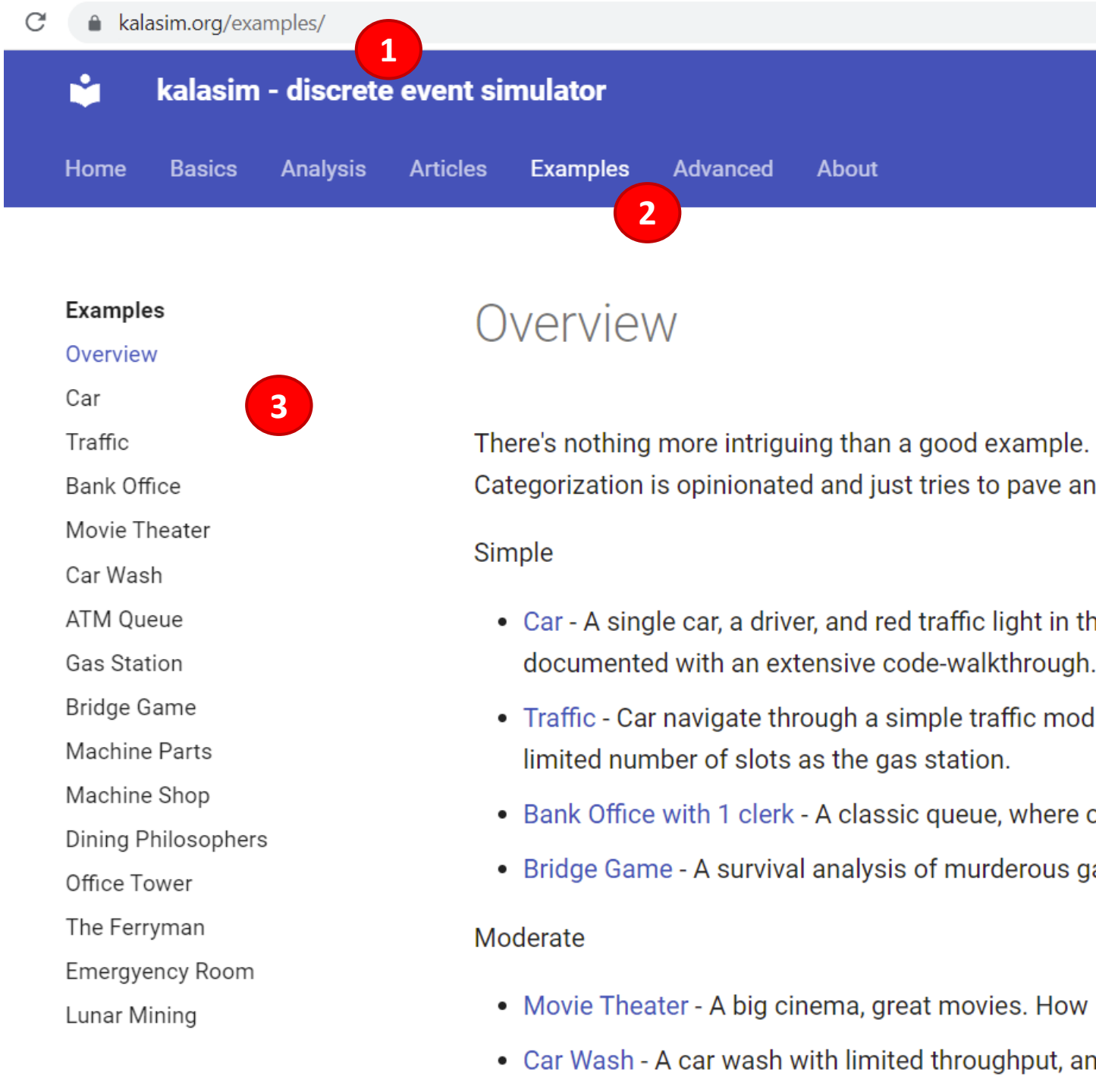
- Hint: Use a `ComponentGenerator` to create customers
- There's a rush-hour between 8 am and 10 am when the arrival exponential time between customer arrivals is just 1 hour

# WHAT'S NEXT?

Project Roadmap (me)

- Evolve Process Animation API
- Environment snapshotting & branching (based on kryo)
- Agent-based modeling
- Evolve process mining toolbox
- (Help others to) Make more money

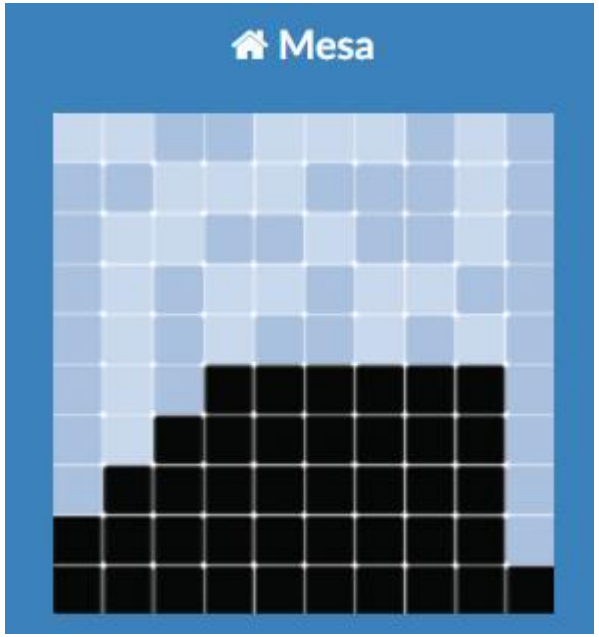
You:



# WHAT ABOUT AGENT BASED MODELLING?

- Grid + step function
- Split step when using parallel scheduler
- Visualization support API

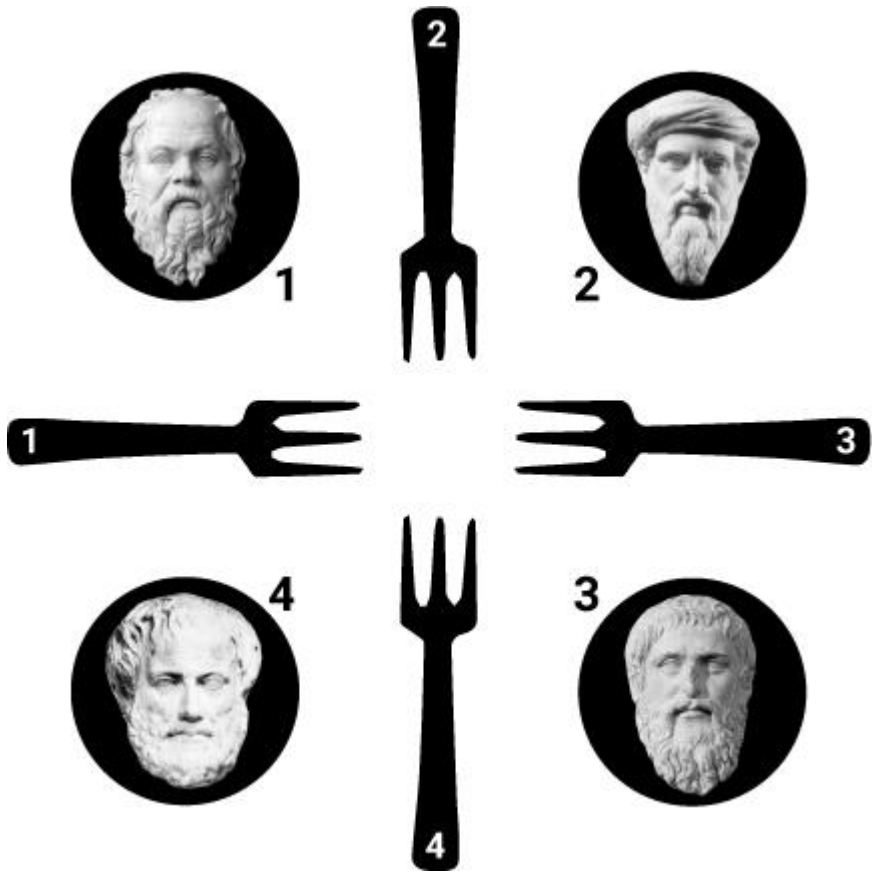
→ Could be easily expressed with kalasim as well




https://mesa.readthedocs.io

LIVE-CODING SESSION

# HAVE FOOD & FUN WITH DINING PHILOSOPHERS



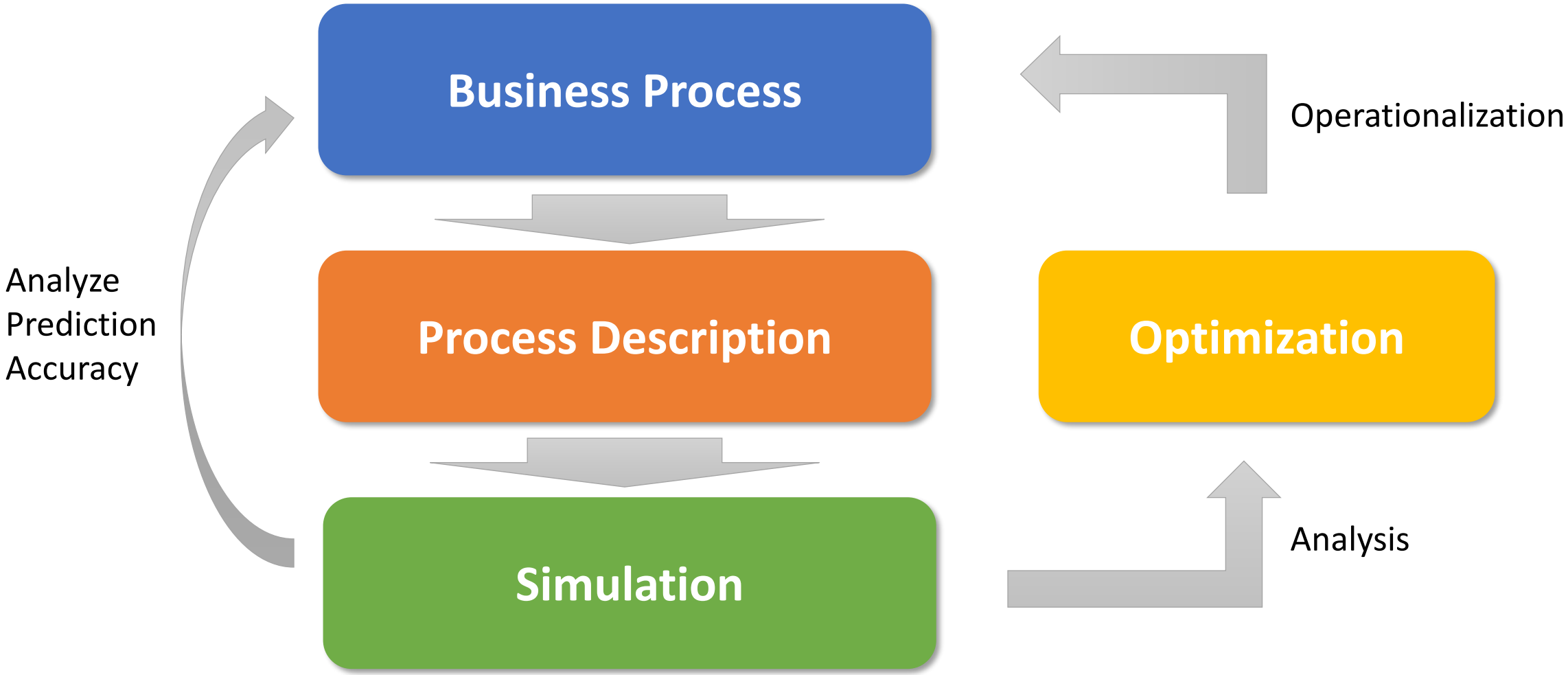

@

CASUS
CENTER FOR ADVANCED
SYSTEMS UNDERSTANDING

https://www.kalasim.org/examples/dining_philosophers/

https://www.youtube.com/watch?v=GzcpJ_7pWvk

# PROCESS OPTIMIZATION LIFECYCLE



# PROCESS MODELLING TOOLKIT - SUMMARY

- Polyglot Experience
- Opinionated View
- Many options → No right or wrong, even if all “experts” have their *way*

(Opinionated) Lecturer Stack

- Data Analysis: R
- Simulation: Kalasim
- Optimization: Optaplanner
- Artificial Intelligence: Rlib/Ray

systema.com/optimization

Blog Resources Support Contact

SYSTEMA
Art of Automation

INDUSTRY 4.0 COMPANY INDUSTRIES CAREER

Digital Transformation

Smart Manufacturing

SYSTEMA Portfolio

SAP Portfolio

Resources

Integration

Automation

Optimization

Visualization

Migration

Optimization is the art of maximizing manufacturing efficiency, throughput, OEE, yield, and quality by monitoring, analyzing, and iteratively tuning manufacturing processes.

# SYSTEMA GmbH
# Art of Automation

“Industrie 4.0” made with ❤

Integration

Automation

Optimization

Visualization

Migration

SYSTEMA
Art of Automation

# JOIN US → CAREER@SYSTEMA.COM

**What we offer:**

- **Permanent contracts**
- **Individual development plans**
- **Individual benefits**
- **Work permit/Visa support**

**Also**

- **Internships**
- **Thesis projects**

- Process Analysis & Modelling
- Scheduling & AI (Anomalies, RL, Regression)
- Factory Physics & Industrial Engineering
- Software Development (Java, Python, C#, SQL,…)
- Big Data (Spark, Kafka, Elastic, Cloud) & BI
- Real-world problems in different manufacturing industries


Headquarter Dresden




# THANK YOU FOR JOINING!

# THANKS TO THE ORGANIZERS INTERNATIONAL WINTER SCHOOL ON SOFTWARE ENGINEERING

# LET'S STAY IN TOUCH!

@holgerbrandl

# IMAGE ATTRIBUTIONS

- The moon (CC BY-SA 3.0) https://en.wikipedia.org/wiki/Moon
- Chip (CC BY-SA 4.0) https://commons.wikimedia.org/wiki/File:Samsung_Galaxy_Tab_2_10.1_-_Texas_Instruments_TWL6032-3960.jpg
- Business Process 1 (CC BY-SA 4.0) https://www.flickr.com/photos/davegray/5630708345
- Business Process 2 (CC BY-ND 2.0) https://commons.wikimedia.org/wiki/File:Gamification-in-business-illustration-web.jpg
- OPENRNDR http://openrndr.org
- Kotlin Features https://github.com/thomasnield/kotlinconf-datascience-talk
- Claude Shannon https://commons.wikimedia.org/wiki/File:Claude_Shannon_1776.jpg
- Ishikawa Fishbone Diagram (CC BY-SA 3.0) https://commons.wikimedia.org/wiki/File:Ishikawa_Fishbone_Diagram.svg
- Money (CC0) https://www.maxpixel.net/Market-Currency-Money-Europe-Euro-Coins-Yellow-1353420
- Diver https://www.flickr.com/photos/scubavagabond/2428241107