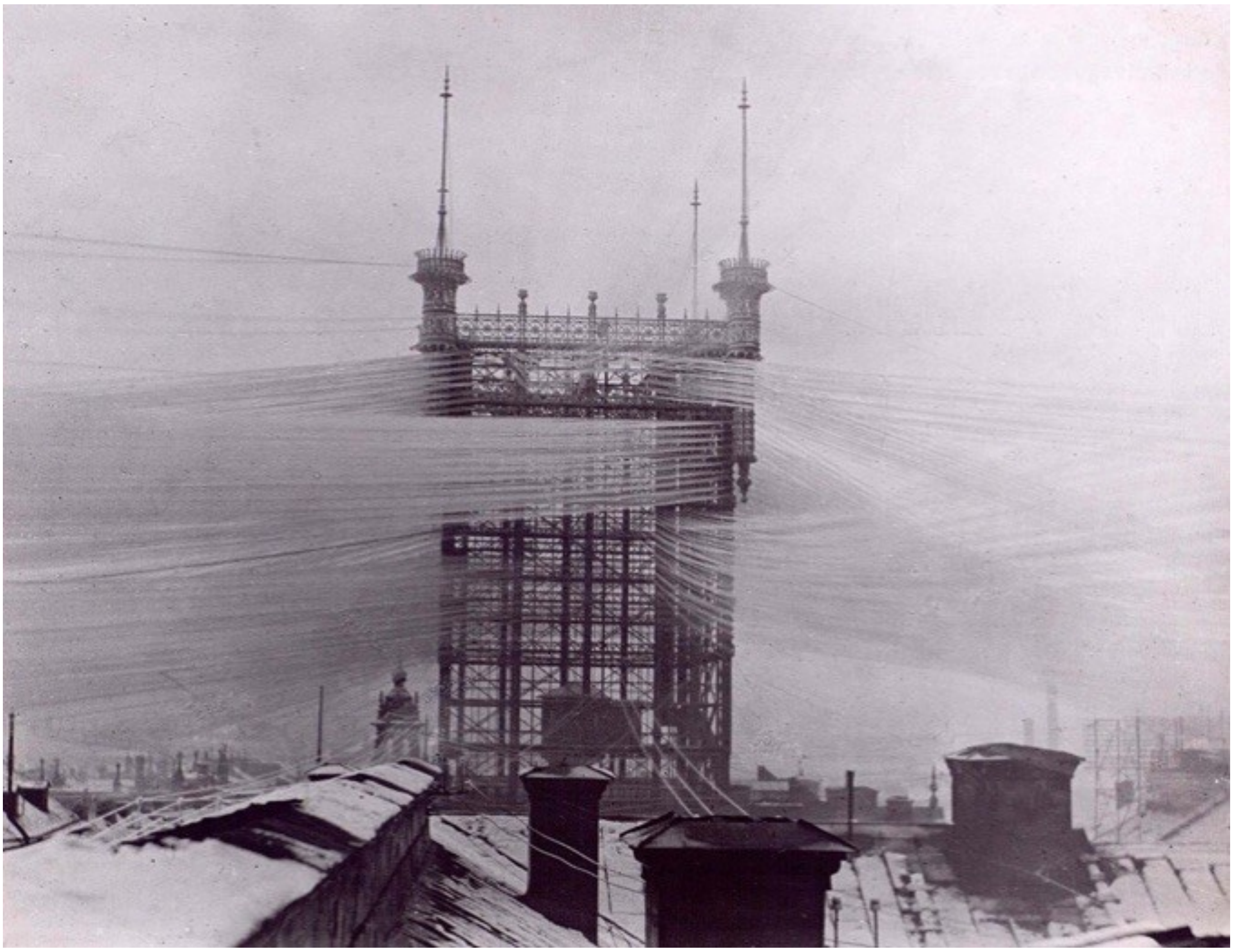
How to use game-changing AI to boost decision quality
Case Study: Maintenance Planning

Tom Savels, Norbert Dolle
Managing Partners at White Space Energy



Disclaimer: The featured photo shows the old telephone tower in Stockholm, ca. 1890. It connects around 5,500 telephone lines. If you're completely comfortable with its complexity, you're reading the wrong article. If, however, it reminds you of your own business and you believe there's always a better way — this article is for you.

In the Intro to this series, we discussed the potential of Game AI for business applications. Businesses generate value through the quality of their decisions. Often, these decisions are very complex. There are millions of choices, conflicting drivers, uncertainty, hidden bias, and so forth. These complexities all add up, which makes good decision making very hard — for humans. In contrast to humans, Game AI is *designed* to handle such complexities. It excels at supporting complex decision making.

To illustrate how Game AI can help to augment decision making, we are publishing a series of case studies. These examples bring out the strengths of Game AI in real-world industry applications.
This article discusses the first case we'd like to cover: *maintenance planning*. We'll focus on the Energy industry, but this is a high-stake problem across many industries.

We will go step-by-step through our approach:

1. Translating the business problem into a game
2. Developing the right Game AI agent to master that game
3. Analyzing the AI's winning strategies — thereby solving the business problem

But, before we get into games, AI and winning strategies, let's take a step back and start with the basics. *What is maintenance planning*, and more importantly: *why is it such a complex problem*?

> "*Industry outsiders might be surprised by how often (…) operators overlook or ignore detailed activity planning, either because more pressing break-fix problems take precedence or because it just seems less interesting in an industry focused on exciting discoveries and new production*." (Bain & Company)

Maintenance Planning — what is it and why is it complex?
Imagine an offshore asset of an energy company. It has ten unmanned platforms that need maintaining. The asset is facing a huge workload of maintenance scope, with about 9,000 activities it needs to execute.

These activities are very different in *hard metrics* such as:

- total number of workers (staff) needed
- skills of the workers (e.g. welders, scaffolders, and technicians)
- material requirements to execute the work
- location of the work
- duration of the work

Also, activities differ in *softer metrics* such as

- relative importance
- relative urgency

Finally, to make matters worse, the execution capacity is constrained. There is a single workboat that transports the maintenance crew to the platforms. That boat has a fixed capacity, limiting the available number of workers on each day and location.

The problem the asset faces is: What is the “best” strategy to plan activities?

This question is complex in many ways.

First, there’s the problem of *scale*. With 9,000 pieces of work, the number of possible planning options is absolutely massive.

Second, there are *constraints*. The workboat can only carry so many workers on any given trip. Crew members each have particular skills and need planning accordingly. Activities may be interdependent, with some depending on the timing of others. Some activities may be executed in parallel while others may not.

Third, softer metrics are typically *opinion driven*. Opinions are often not quantified and biased. Priorities are driven by “whoever shouts the loudest”. Different stakeholders have different opinions. And these opinions are competing for the same resources.

Finally, *conditions change* all the time. Inspection results, production upsets or leaks frequently change the portfolio. So, on top of being very large, constrained and opinion-driven, the work scope also changes all the time.

So, it’s fair to say that the asset is facing a complex landscape of drivers, constraints, and opinions — and math. Finding the “best” strategy to plan work is not straightforward at all.

The *existing* process to handle this challenge is based on manual work. Plans are generated through a combination of Excel, experience and gut feel. The result is a far lower than expected efficiency of the maintenance crew. Plans are unstable and maintenance backlog increases year-on-year. This backlog is affecting the safety and production performance of the asset.

“*Picture a Friday night after work in any oil town, from Aberdeen to Calgary or Perth. Drillers and maintenance engineers recount the week’s small victories — not a few of which include boasts of heroic efforts requiring long hours of overtime to fix unexpected problems, or premium prices paid to rush parts and people to where they were urgently needed — all part of the “can do” spirit for which the oil and gas industry is famous. While these make for good stories, this is no way to run a business*.” (Bain & Company)

In other words, improving planning efficiency is a critical problem to resolve. And current processes don’t cut it.

Fortunately, there’s an alternative: building a game.

Game AI — Step 1: Translating the problem into a game

Our first step is to take the business problem at hand and turn it into a game. For Game AI to be effective, the accuracy of this translation is critical. In practice, this step requires an in-depth business-side understanding of the problem.

Like real-life games, all business games have the same key elements that define them. There's a goal, players, moves and rules, and a scoring system. Let's go through these elements for our maintenance game.

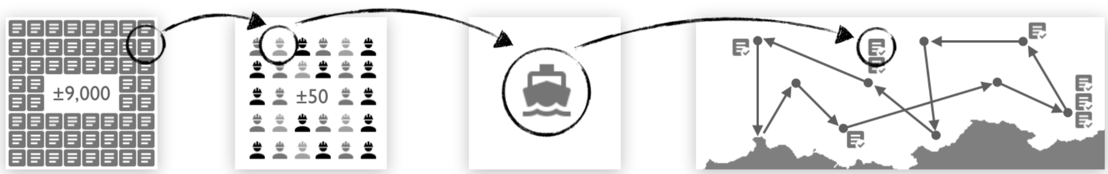
Goal of the game

The *goal* of the game is to assign resources to tasks in time, in the best way possible. *Resources* are the workboat and the maintenance crew on the boat. *Tasks* are activities taken from the (ever-changing) portfolio of work scope.

Players, moves & rules

Maintenance planning is a *single-player* game, with the AI agent competing against itself. The *moves* that the agent makes represent planning decisions. They are:

1. Which locations will the boat visit, and in what sequence?
2. How long will the boat stay at each location?
3. What crew is onboard? What activities will the crew complete during each visit?



These decisions will need to adhere to various rules such as

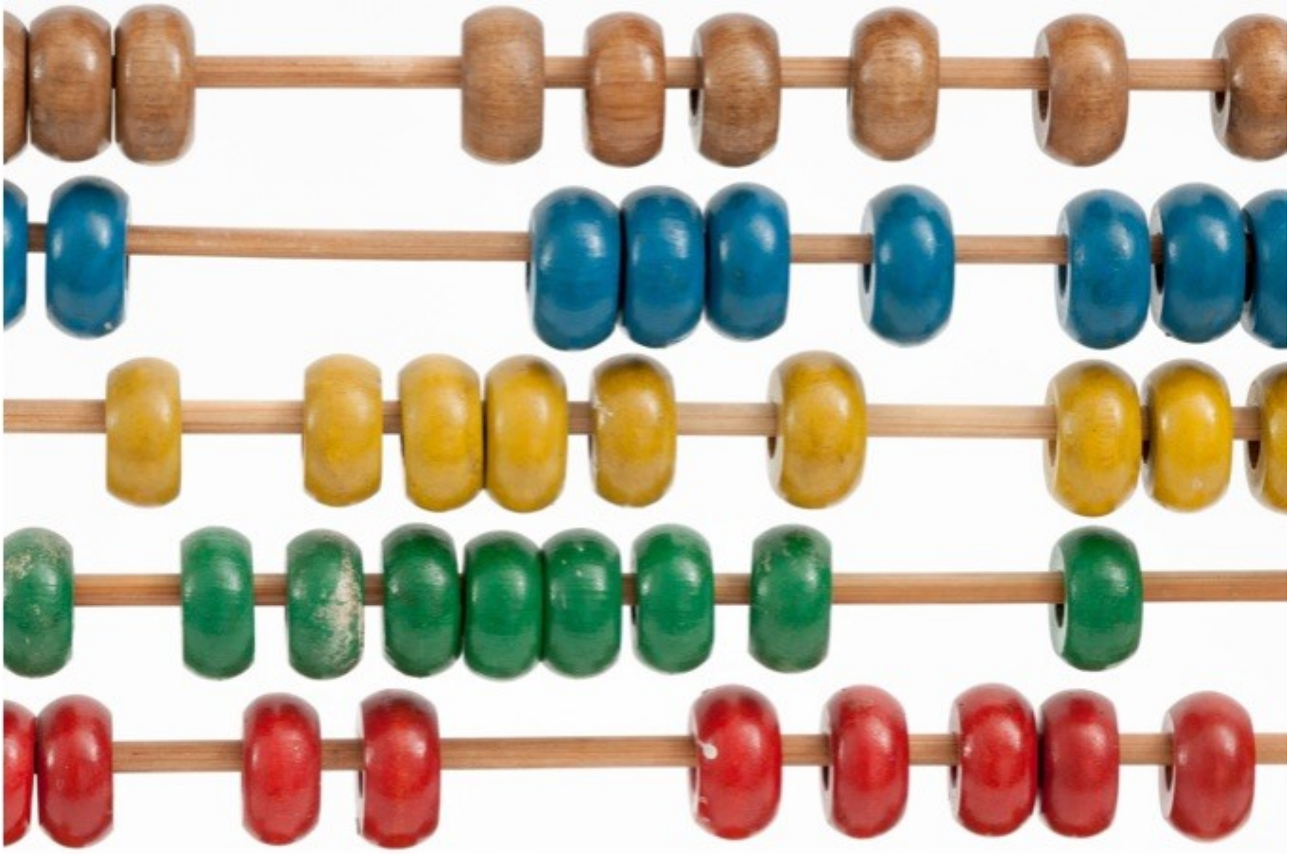
- The workboat can only transport up to 50 workers.
- The workboat needs to return to shore every 2 weeks for a crew change.
- The workboat can only support one platform per day
- Working hours on the platform are 8 AM to 6 PM during summertime, and 9 AM to 5 PM during wintertime.

Scoring system

With rules in place, we have a player that can make legal moves in a planning game. Stringing moves together corresponds to a maintenance plan. *Mastering* the game means learning how to make *winning* moves. And a sequence of winning moves results in an *optimized* plan.

To determine how to win, we need to introduce a *scoring system*. The scoring system reflects the value of each move a player makes. That is, each planned maintenance activity — a game move — earns the player points. The number of points reflects the value of executing a maintenance activity at a given time.

Of course, there will be differences in opinion about this valuation. That is, different opinions will give rise to different scoring systems. For example, assume the production manager prioritizes work on the highest producing platforms. We can accommodate this with an option to earn extra points for work on those platforms. Or, let's say the maintenance manager prioritizes reduction of backlog. We can accommodate this by rewarding work on backlog items. Every set of opinions corresponds to a set of scores, and a Game to play. And every Game that is mastered corresponds to a plan optimized for the opinions that drive it.



At this stage, all main components of the maintenance game are in place. Next, let's explain how we develop the AI to master it.

Game AI — Step 2: Developing the right AI agent to master the game

Game AI as a research topic has seen a huge boost quite recently. There's a wealth of papers on cutting-edge solver architectures. There's also quite some open source code shared by DeepMind, OpenAI, and others. This is great, but it doesn't mean that developing an AI agent for a new game — like the maintenance game — is trivial. Matching a game with the best algorithm(s) can get quite tricky. Development is often iterative, with the game and the solver being tuned in parallel.

In general terms, we use a combination of

- Deep Reinforcement Learning (DRL),
- Tree Searches, sometimes stochastic,
- Operations Research (OR) approaches, and
- Evolutionary Algorithms (EA)

This list is not exhaustive but gives a feel for the breadth of architectures we look into. A couple of things are important to us when developing our AI:

- *Performance* — how good is it in making decisions?
- *Speed* — how fast does it learn?
- *Scalability* — how many options can it handle?
- *Ease of use* — how much tuning does it need to make it learn?
- *Robustness* — how sensitive is it?

- *Fit with the game* — is it simple enough? We don't want to come up with a sledgehammer to crack a nut.

In the case of the maintenance game, we observed the following:

- Tree Searches have potential but our option space is too large to use them stand-alone.
- OR approaches are very performant at small scale but break down at our scale of interest.
- Deep Reinforcement Learning is flexible but needs careful tuning to stimulate learning.
- Evolutionary Algorithms don't fit this type of sequencing game very well.

Approach	Pro	Con
(Monte Carlo) Tree Searches	Good at incorporating sparse feedback	Balancing tree size growth vs exploration is tricky
(Deep) Reinforcement Learning	Very flexible Can handle large state/action spaces	Needs lots of care when dealing with sparse feedback Fine-tuning may require lots of TLC
Operations Research	Can be very performant - if applicable	Very inflexible Only applicable for very specific problems
Evolutionary Algorithms	Flexible	Doesn't naturally fit with the mechanics of the maintenance game

Like ordinary games, *business* games span a huge spectrum in setup and complexity. As such, Game AI agents tend to be very different as well. A one-size-fits-all solver architecture approach doesn't work.

So, we build our AI agents using elements of many architectures. Just like AlphaZero combines Reinforcement Learning with Tree Searches to master Go.

For this Case Study, we developed a hybrid approach as well. At the *top level*, we used Tree Searches and Deep Reinforcement Learning. These approaches excel at making large-scale decisions (e.g., where do I send the boat and for how long?). At the *bottom level*, we built an Operations Research engine. It excels at dedicated sequencing (e.g., what work do I execute at any given location and duration?). The combined architecture is fast, flexible and performant. By combining the strengths of its building blocks, our solution outperforms single-engine solvers.

Game AI —Step 3: Analyzing the AI's winning strategies — and solving the business problem
Quality decisions need a complete overview of all viable alternatives. And an understanding of the trade-offs between these choices. Through our game setup, it's easy to translate different opinions into alternative plans.

For example, one may argue that 24 working hours per day is possible on some platforms. And (this is an *opinion*) that this will massively increase productivity. We can then update the rules of the Game, let the AI bot master this new game, and review the resulting optimized plan.

Another example: the maintenance manager may feel strongly about his backlog. He may argue (another *opinion*) that reducing that backlog is of the highest priority. We can reflect this in the scoring system and let the AI agent play and master the game again. This creates another alternative and optimized plan.

The beauty of the Game AI approach is that any number of opinions can be compared using the same basis and metrics. For every set of rules, the AI agent produces an optimized plan. Comparing optimized plans is comparing apples to apples.

So, how then do we compare these alternative plans?

First, we define *key metrics* with our business stakeholders. These reflect indicators that are important to them and drive decision making. Sometimes these are existing metrics, sometimes we define new ones.

In this case study we used the following metrics:

- % of safety-critical scope completed
- size of the backlog at the end of the year (# of work scopes)
- production volume at risk
- total number of work scopes executed

Second, we define the *scenario's* that need evaluating. These represent sets of optimized games to be played — as many as needed, up to thousands. The scenario's and their metrics give full insight in trade-offs between opinions and resulting plans. This then allows the ultimate decision-maker to make an informed quality decision.

In this case study, we found that up to 30% more work can be planned compared to the existing planning philosophy. Normal business improvement initiatives rarely lead to double-digit improvements. In addition, in the Energy industry double digits represent huge monetary value.
A 30% improvement is *huge* and allows for a reversal of the year-on-year increasing trend in backlog. But, is 30% a surprise? No, not when you think about the outdated toolset that the team is using to play such a complex game. Our improvement is — at least partially — driven by the current low standard. But that standard is real — it's what planners face every day. They shouldn't have to.

Replace your outdated toolset, and redesign your processes

In our Intro, we recommended taking your complex business problems to the drawing board. We urged you to replace your ancient Excel planning sheets with 21st-century tools. To challenge yourself and to re-design and improve processes and ways of working. And re-design them around the latest technology.

We hope that this case study is a motivation for change. Most complex business decisions are based on flawed or incomplete analyses. If you look at your own business and didn't know how to improve, then we hope this article has sparked ideas.

We are very interested to hear your views. Feel free to reach out on info@whitespaceenergy.eu.