HOW DOES PREACTOR RESPOND TO THESE KEY QUESTIONS

1 **What makes Preactor flexible**
Unlike ERPs, Preactor is built to adapt each companies data and buiness rules. This means that the database, calculations, objects and views are fully customized to each companys requirements. At the same time Preactor already contains plenty ready to use tables, views and objects so that companies do not start from scratch and allow for first results to be tested in weeks.
Speed and automation tools contained in Preactor also serve this purpose because it makes results evident and easy to communicate

2 **What makes Preactor intuitive and user friendly**
All interaction of the planner with the tools is through graphical interfaces: colored boxes, calendars, grids and alerts are all interactive and graphical which allows plnners to view and edit all variables from only one place and to have immediate alerts regarding any affected variables
By double clicking on any object, full order details are displayed containg all relevant variables of each operation, whic can be manually edited or downloaded/uploaded through Excel files

3 **Which data comes from ERP and which data comes from Preactor?**
Typically, ERPs handle demand order entry, bill of materials and inventory. Preactor usually handles routes, constraints, calendars, setup matrices, kpis and floor.
MRP may be solved by ERP or Preactor depending on specific needs and the detail of pegging rules required

4 **How are options, routes and setups configured easily in Preactor?**
Routes are modelled in Preactor using attribute rules at planning level data. By setting up automatic routing configurations, we avoid having to manually create individual routes for each product and instead we make Preactor produce this information for us automatically.

5 **How can the planner interact with Preactor planning board and other objects to make changes on the schedule logic and priorities?**
Gantt chart allows the planner to move operations using a drag drop interface
Operations also contain a priority field which can de updated manually or using an Excel file. Priority tells Preactor which orders go first
Grid object allows the planner to move schedule table rows up and down to establish a higher or lower start time
Algorithm selection and calibration produce different rules and solutions in the board
Filters and highlighting tools allow the planner to unschedule or reschedule portions of the plan
External Excel files and systems can also be fed into Preactor to enforce a specific sequence on each resource

6 **How does Preactor classify and identify problems ? What are the standard problems that Preactor identifies and how does it fix them?**
Constraints and limits are classified within Preactor as either "hard" or "soft". Hard constraints are forbidden, and therefore the system will not allow such limits to be violated
Soft constraints are monitored and warned as undesired, but not forbidden.
Examples of hard constraints or invalid movements include errors on the operations sequence or use of locked resource calendars.
The system prevents the user from making these manouvers
Problems such as idle capacity, suboptimal use of resources or scheduling orders before material availability can be modelled as "hard" or "soft" constraints and may therefore be reported and fixed, but not necessarily forbidden.
Preactor requires the user to establish acceptability limits on each relevant variable to monitor and enforce them accordingly.
The system will then classify and highlight the main problem on each order based on the lmiits and offer customized corrective actions to the planner.
The planner can then choose to fix the issue at hand or not

7 **Which relevant outputs, views and information extractions does Preactor produce for work areas?**
Gantt Planning Board: Preactor main interface allows planner to interact manually with orders and resources to drag and drop the orders and run algorithms
Planning Grid: Grid view that allows users to read and reaccommodate schedule rows
Schedule Reports: Preactor generates and stores schedules to communicate to the plant the start and end time of each process on each machine
Scenario Comparison Charts: Charts to view side by side the plan vs the real on each operation
Plot Charts: Secondary constraint levels and usages driven by the schedule
MPS Inventory Projection: Projected inventory levels for each component over time
Material Tree View: MRP view to display bottlenecks and shortages on all material levels for each order
Action List Panel: Customized actions built to automate multiple tasks
Performance KPIs: Continuous updated KPI performance metrics to monitor plan vs real performance and accumulated productions to date
Shop Floor Execution Panel: Event interface to allow each machine to report their status, progress and relevant event such as brakdowns and material consumption

8 **How do I know Preactor is correcting and solving the schedule to maintain the best possible solution based on my priorities?**
Scheduling problems are complex large scale problems that involve the interaction of multiple variables and rules.
Planning goals are usually in conflict with each other (eg. We want less inventory and faster delivery, we need higer product mix and less setups) and therefore planning algorithms are typically set up to prioritze some key objectives first and to accomodate the rest when possible.
KPI metrics monitor the deisred level vs the planned level obtained by algorithm to meet the planners needs. To determine how good a specific solution is, the planner can manually intervene to edit the solved plan and test o to see if his manually calculated plan has a better (and valid) performance over the suggested solution. If the tools os properly configured, any manual attempts from the planer to improve the performance should result in a diminished performance in a few of the relevant metrics

9 **How and where are external events fed into Preactor and how do they change the schedule? Whcih are the usual events that affect the schedule?**
Standard external events include changes in demand, changes in material supply, shop floor events.
Demand are usually external as a client changes an order priority , duedate or quantity. These changes are normally communicated from sales tables in the RP into Preactor
Material supply changes include events such as change in delivery date or loss of quality in inventory.
Preactor responds by reallocating material pegs, splitting productions or launching leveling material orders.
floor events include machine breakdowns and waste of materials and slow progress. These eventas are usually launched from floor panels at each machine
Allocation manouvers- Planners can decide to change order priorities, switch an algortihm or lock resource availability.
These manouvers affect the schedule plan and the subsequent resource allocations to it
Resource calendars Resource availability such as labor attendance or incorporating a new machine make Preactor rebalance resource loads

10 **How do I know a change has ocurred and how do I know how it has affected the schedule?**
Changes are recorded into a log every time the planner hits the "SAVE" button. The system then compares the old and the new and generates a change log view

11 **When a change creates a problem in the schedule, does Preactor fix it automatically or can I intervene to fix it?**
Automated or manual repairs are optional. Preactor includes an automation tool called PESP which alows the system to be configured to execute scripts automatically based on certain events or triggers. The before and after logs measure not only the schedule positions but also the resource or tolerance levels of any relevant metric to be monitored so before and after levels can be traced.

12 **How does Preactor compare scenarios from different algorithms? w does Preactor compare plan vs real? Initial plan vs latest plan?**
Scenarios from different algorithms can also be saved and compared side by side or through performance metrics to evaluate them. Just as plan 1 solved by using

algorithm X can be saved and compared to plan 2 which was solved by using algorithm Y, plan changes over time and real execution data is stored and considerd as a plan or scenario to be compared with the first

13 **Which planning techniques and technologies are used by Preactor to solve problems?**
Preactor contains mosty heuristic rules that can be customized and created from scratch to produce schedules. As planning algorithm developers we have also incorporated optimization techniques such as linear programming, nonlinear programming, combinatorial optimization, genetic algorithms, machine learning, critical path planning and constraint based scheduling a-mong others to solve specific planning and scheduling problems and theory of constraints applications. Among applied technologies, languages and math solvers such as LINGO (LINDO), PYTHON, R, AMPL (CPLEX) and Preactor simulation API can be applied.

14 **Which programming languages are used in Preactor and what are they used for**
Manip: A shell language to configure high level scripts and filters to configure the system rapidly
Preactor .NET API: Preactor object layer to program and manipulate rules and data flows using customized actions
Math Solver: Used to model math optimization models when required
SQL: Access to data layers is open which allows users to customize queries, reports and views and to introduce new tables and trigger commands
Web languages: Used to customize platform and apps on web platform environment
SCADA languages: Used to configure sensor level communications with the floor

15 **Which planning algorithms are contained in Preactor and how do they work**
Forwards: Look for the soonest available resource for an order and schedule it asap
Backwards: Start backwards for the duedate of the last process to deliver and look for the last possible resource to process as JIT
Bottleneck bidirectional: Isolate the bottleneck process of each order first and optimize their load. Then schedule remaining processes fwds and bcwks respectively
Wip Minimization: Peforma double pas son each order to schedule fwds first and then backwds once thehighest waiting process is found
Setup Minimization Group siilar orders togetherand balance the capacity with the grouped orders without violating delivery
Parallel Loading: Focus on making the most similar possible load in a set of resources
Hybrid Callibration: Usally a balanced blend of the previous goals and techniques, running multiple passes and reacommodations
Custom algorithms: Personalized algorithms and optimization models for each specific industry

16 **What are examples of custom planning algorithms you have implemented and what did they achieve**
Mold balancing in Crocs
Expense minimization in Maxion
Reverse Logistics in Corona
Inventory optimization for Polykon
Pattern cutting and allocation in CCL

17 How does Preactor reduce manual data entry?
By usig automation, mass reacommodation manouvers and validation rules, Preactor allows users to update masisve data through the use of a few clicks. Preactor creates linked events so that when one item is changed or moved, adjascent orders and events are updated coherently.
The systems automated algorithms and manouvers continuously repair and adjust impacts on materials and capacity
Finally, Preactor implementations normaly incorporate the use of sensors, rfids and mobile tools so manual entry is drastically reduced.

18 How does Preactor synchronize plan vs real actions accross systems?
Communication between Preactor and execution systems can be automated to carry Preactors manouvers into the ERP and viceversa.
This is accomplished accessing the ERPs communication layer objects to automate log activity extraction from and to ERP systems

19 Which tools can I use to produce reports or extract data form Preactor
Preactor reporting tool: Preactor includes its own interactive report builder wich writes xml files that can be written to Excel or pdf
Preactor Interfacing tools: Prector interfacikng scripts allow users to customize file layouts for importing or exporting data
Preactor web platform: KPI views and reportas are published to web platform using HTML mappings
Preactor database access: Standard SQL queries can be structured to Preactors database

20 **Whats the difference between data kpis, performance kpis and customer kpis?**
Data kpis are used to monitor the data quality of all relevant orders to be planned. Missing data or errors is detected and communicated to the process owner for review.
Performance kpi are used to monitor the achieved goals in the plan through the application of rules and parameters for the problem at hand. Standard performance kpi metrics include throughput level, late orders, idle time and inventory goals. Customer kpis are metrics ony relevant for the customer.

21 **How does Preactor handle multiplants**
Plant detailed models are built for each factory and linked to a central, less detailed planning model. The central model builds boundaries or requirements for the other factories and as plants build their detailed schedules informaton is shared and adjusted

22 **What is Preactor implementation methodology?**
We have a standard work methodology that flows through multiple milestone and stages. The work plan is published in our project tracking platform KiMI and files and examples of all work is submitted for review through the tool.

23 **How does Preactor attain each of the improvements it claims to achieve? How can I testify to such imkprovements in my company?**
As plans start to be tested and published in a Preactor implementation, a snapshot of the initial metrics of the company are recorded. As implementation progreses a kpi on each of the relevant mertrics is tracked to evaluate its performance over time before and after Preactor is implemented. Customized planning algorithms that compare thousands of feasible solutions and automated data flows attain the main goals of improving solve time, increased throughput and reduced wip inventory.
The mrp and materials manouvers normally achieve the reduction in material waste, and the use of the shop floor sensors and tools drive the improvement for throughput and consistency in the execution.