Case study

Chapter 2

**2.16 Case Study**

In Chapter 1, we saw the introduction of the project of our SaaS vendor. We will now continue

with our case study and discuss how the project initiation part of the case study was done for the

release 6.0 of the product of our SaaS software vendor.

***2.16.1 Project Charter***

In the industry there are no good solutions available for appointment scheduling. By creating this

functionality, the SaaS vendor wants to become a leader in this arena. Existing as well as potential

customers are also eagerly waiting for a good solution that could substantially cut the waiting time during the shipping–transporting–receiving operations of goods. After building and implementing

the appointment scheduling solution, the SaaS vendor will be able to effectively satiate the

needs of its customers.

***2.16.2 Project Scope***

Appointment scheduling functionality is the biggest feature for the 6.0 release. It could not be completed in one iteration. So, it was divided among four iterations. Calendars were created separately for dock doors, warehouses, organization, etc. in the first iteration (release 5.3).

the existing functionality of appointment scheduling up to the release of 5.2 was limited only as a mechanism to announce the arrival of the truck for either shipping or receiving at the ware- houses. It was not actually making an appointment, as no constraints were considered for making an appointment.

For truck appointments at dock doors for loading/unloading, there are a lot of constraints to be considered. \therefore, it was decided to create the functionality over many phases. thus, in release 5.3, only constraints of truck type and goods types were considered for making appointments. At this juncture, calendars were also used for dock doors (if for instance, a dock door is open from 6‚ to 6 „‚ on Mondays, then if any truck arrives after 6 „‚ its appointment will be considered only for the next day). In release 5.5, the functionality was enlarged to consider constraints of labor avail- ability, quality control inspector availability, expected arrival time of truck, time window for making an appointment, and dock door type. In release 5.8, the functionality was enlarged to consider constraints of business partner preference, truck capacity, reservation frequency on a dock door, and reservation lead time. In the final iteration, the functionality was enlarged to consider the constraint of time gap between appointments. In this iteration, more time was given for testing than for development as the vendor wanted to make sure that all the functionalities work well, and that the software product do not fail. Instead of having a large number of poorly made features, it is better to have a software product with a limited number of features that are robust and will not fail. Reliability was the top priority.

After the four iterations and the entire 6.0 release, the software product should be able to be implemented with the new functionality by all existing customers as well as new customers who

will sign contracts during this period.

***2.16.3 Project Objectives***

¡e software vendor could see that there was a large gap in the market for supply chain management

software solutions in the grocery retail segment. The software vendor’s flagship software

product already had functionality for transportation management, inbound logistics, outbound

logistics, fleet management, transportation rate calculation management etc., up to release 5.6 of

the software product. ¡e project charter for the project to release 6.0 (through minor releases of

5.3, 5.5, and 5.8) was to create additional functionality, such as appointment scheduling for warehouses

with incoming trucks and an audit trail for all transactions.

Most software planning systems use complex logic to implement solutions which could be used in real world planning systems. Unfortunately, most of the systems fail miserably in delivering on promises. One reason is that real world happenings are far from ideal. There is always some unplanned risk lurking around the corner that can upset the rhythm of even the most meticulously planned activities. then the planning logic is error prone. A good algorithm was needed in the first place. Secondly, it had to be implemented in such away that it would provide a real world solution. So, it was decided to go in for hard as well as soft constraints for making the appointment scheduling of an incoming truck for loading/unloading at a dock door. ¡e soft constraints could be overridden if some other constraint that is higher in hierarchy is satisfied in the current situation. But the hard constraints are such that they will never be overridden. All the constraints are thus put in a hierarchy, with some of the constraints higher up in the hierarchy and others lower.