HNU HOCHSCHULE NEU-ULM
UNIVERSITY
OF APPLIED SCIENCES

Beer Game Implementation using Raspberry Pi's to verify supply network and analysis

Subject: BI-Strategy by Prof. Dr. Engel
Master of Business Intelligence and Analytics Winter term 2018/19
Neu-Ulm University of Applied Sciences

Presented by:	Emmanuel Osei-Oppong	BIA	262583
	Mayuri Kalegowda	BIA	264413
	Krishna Prasad Wagle	BIA	265632
	Leonie Behrens	IEIM	264697
	Wolfgang Schwarz	IEIM	264396
	Yaroslava Zhuk	IEIM	232790

Table of contents

1. Introduction

In the last few years the interest in logistics and supply chain management has grown explosively. This process has led many companies to evaluate their supply chains. However, mostly, this has been done based on experience and intuition; very few analytical models or design tools have been used in this process. Meanwhile, in the last two decades, the academic community has developed various models and tools to assist with the management of the supply chain and one of them is the Beer Game.

In its simplest sense, the supply chain is nothing more than a holistic view of a business enterprise from the origin of raw materials through to the use of the completed product by the ultimate consumer. That process will likely involve multiple companies and may cross several industries. In the traditional methods of operation, each of those companies and industries will operate independently and in isolation from each other. Oftentimes, they will treat each other as adversaries. Communication is limited to only the necessary transactional information required to create and fill orders from one company to and from the next link in the chain. Rarely is the information complete enough to offer any real opportunities for planning. There is a complete reliance on forecasting, which is notoriously inaccurate. In fact, the further back in the supply chain a company is, the less accurate the forecasting information will be. The root cause of all this is that links of the traditional supply chain are considered independent rather than as parts of a single system.

2. BI Strategy for our Beer Company

Business Intelligence Strategy is developed from a firm strategy so that it can be aligned with the strategic direction of the firm. Considering this we developed strategy for our Beer Company as below

Our Beer Company Strategy or goal: Our Vision is to brew quality drinks for all everywhere

Mission: To produce quality beers to satisfy customers in a safe, reliable and environment friendly manner through modern methods

With our vision to brew quality drinks for all everywhere we ensure our supply chain management team is also in line with the vision.

Beer Supply Chain Management Team will support the Company's business strategy by developing and managing processes in procurement, logistics and value analysis. This enable us to manufacture products at the highest quality and ensure their availability at the lowest total supply chain cost while supporting our operations, our vendors and our customers.

The Strategic objective of the Supply Chain using Balance scorecard:

The balanced scorecard is a conceptual framework for translating an organization's strategic objectives into a set of performance indicators distributed among four perspectives: Financial, Customer, Internal Business Processes, and Learning and Growth- Supplier

Perspective	Strategic Objective	Measure/KPI
Finance	1. Reduce Supply Chain Cost 2. Lower Inventories	Inbound vs Order Outbound vs Order
Customer	1.High Sales Forecast Accuracy 2. Satisfied Customers	Orders Received vs production
Internal Business Process	**1.** Fast processes 2. Reduce process costs 3. High product quality	Production time per round
Learning and Growth - Supplier	1. Increase product availability **2.** Reduce transaction costs	Production per team per round

Fig.1: Balancescore card

Business Intelligence Strategy for our Beer Company

Our goal for BI is to assist the company to achieve its objective by providing accurate and timely information for better performance management and decision-making.

We are going to achieve this BI Strategy by

- Treating Data like Oil.

Our data strategy is to ensure that Data captured are accurate and clean. This is to avoid the problem of Garbage in Garbage Out situation. Assist in the operational implementation of process so as to generate enough data internally. Human manual input will be reduced to the barest minimum.

- Segregation of Report with simplified Dashboards.

Dashboards are designed based on the KPI set in the strategy document above.

Administrator who is acting as the CEO will have on oversight of the overall performance of the Company with the capability of filtering the report to specifics. The other users will only see their sector performance by the KPI set.

- Continuous training of staff

Training of various staff to be able to meet the self-service target of their unit by providing an easy to use tool.

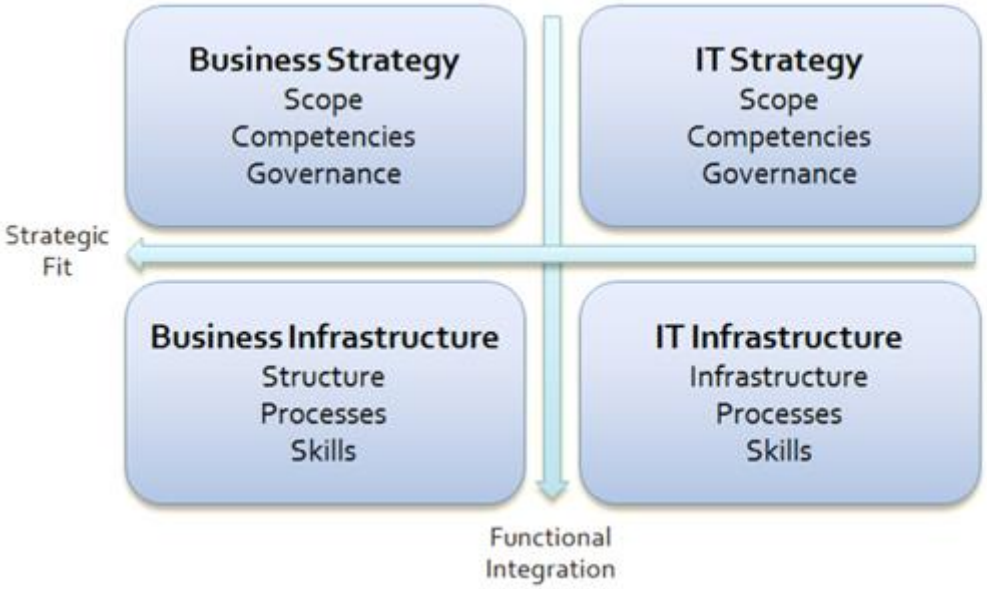

Fig.2: SAM Model

Our BI strategy was developed based on the Strategic Alignment Model (SAM) as shown in figure which is defined in terms of four fundamental domains of strategic choice: business strategy, Business Intelligent strategy, Organizational Infrastructure and processes, and Business Intelligence infrastructure and processes. We define strategic alignment as a continuous process which involve interrelating the coherent combination of the four fundamental domains: business strategy, BI strategy, organizational infrastructure and process, and BI infrastructure and process in order to contribute to the organization's performance. Based on this BI strategy we have implemented the Beer game using raspberry pi's and verified the results by analysing and visualising the data obtained by running the game as explained in the following sections.

3. The Beer Game

The following chapter deals with the basics and the rules of the beer game. This is a physical simulation game that allows individuals who play it to experience a simplified multi-echelon production-distribution system using poker chips signifying inventory, preprinted playing cards indicating the fluctuations in demand at the retailer, and small forms on which the players write their orders to their suppliers. In this physical simulation, the players find themselves immersed in a simulated environment that reflects real-life conditions and, at the same time, is simplified in such a way that it can be seen in its entirety by looking around the table. The physical format of the Beer Game is the most commonly used format. Furthermore, many experts think that it is the only format that truly allows the participants to experience all the features that a real system has to offer . The scenario is that the retailer who sells beer is served by a distributor. The distributor is served by a warehouse, which is served by the manufacturer. Therefore, there are only four elements in this simple supply chain. There is a four-day lead time between each of the elements.

The current version represents a four-stage system that includes a retailer, a wholesaler, a distributor, and a factory. This version includes two-week delays both in the ordering process as in the shipment of goods down the pipeline. Additionally, the system has a one-week delay in the ordering side of the factory end and a two-week delay in production. The factory has unlimited capacity and unlimited access to raw materials. Typically, Beer Game sessions run for three to four hours including the debriefing of the game.
The general aim of the Beer Game is to shorten the total cost for everyone in the supply chain by maintaining low stocks but nevertheless managing to deliver all orders. It is in other words not the intention that you should try to reduce your costs at the expense of the other players. The game has maintained its structure and general mechanisms of play for more than 30 years and it is unlikely that it will change. The most criticized problem with the game is that 'it takes too long to play', and that 'it is very confusing'. However, these characteristics seem to be present in real life in such a way that the players and facilitators keep on finding the experience very rewarding and informative. It seems to be that; the Achilles heel of the game has to do with how to communicate the powerful insights that the game has to offer. Forrester expresses it by saying "you want to be able to communicate the insights that the game has to offer, create great emotional involvement on the part of participants, and at the same time, hopefully, not teach anything that is incorrect in the process". It seems to be that "what is important is not the game, but the gaming experience" and the facilitator plays a key role in generating an adequate gaming experience with the Beer Game. Additional effort to enhance the current support materials for debriefing the game is warranted. To build on the success of the Beer Game as a vehicle to disseminate system dynamics concepts, the development of new games is a promising avenue for research and development within the system dynamics community.

4. Implementation

The main goal of the work is to present an outline of the theoretical background of the beer game and indicative development of a computerized version of the beer game. In the following paragraphs,the implementation peculiarities are explained more precisely, beginning from the technical basics, the description of the used Google Spreadsheets, followed by the description of the implemented procedures and the handling of the software for the user.

General Process

The core process of the Beer Game consists of ordering, producing and delivering the *products* Blue, Purple, Red and Yellow. For understanding purposes, one can imagine the first player being the Brewery, that purchases material from a place of unlimited resources. Player 2 and Player 3 can understand themselves as wholesalers, while the fourth and last player could be the retailer, who receives his orders from a customer.
The *place of unlimited resources* is technically a piece of code, that simply returns the value as a delivery, which was entered as an order previously from the first player.
The *customer* orders essentially with values, that are pre-defined from the admin and therefore can be random, arbitrary and hard to plan (like in real life).

Both orders and deliveries are being executed on the screen of the RPi, but technically handled differently.
The first step is to receive the ordered orders via an NFC tag that symbolizes actual, physical goods. For that, the NFC tag with the delivery needs to be hold right onto the NFC reader in the beginning of the round. The inbound warehouses for each product are being updated and the player can start producing the desired goods. Once the production has succeeded, inbound warehouses will be decreased, whilst outbound warehouses will be increased (because the raw material was transformed to finished goods). Now the player will see himself in the position to fulfill the order, that was previously placed to him from the next player. The respective amounts need to be entered on the RPi, the NFC tag placed on the connected NFC reader and the "delivery" needs to be initiated via a click on the respective button. After that, the player can decide whether he needs to order new products from the previous player. If yes, he can enter the quantities on his RPi. By clicking on "order", the order will automatically be transmitted to the previous player via the common network of all devices.
Now the NFC tag is handed over to the next player. This procedure repeats every round. All data (Order, Production, Delivery) is stored into the database.
The general process is visualized in the illustration below.

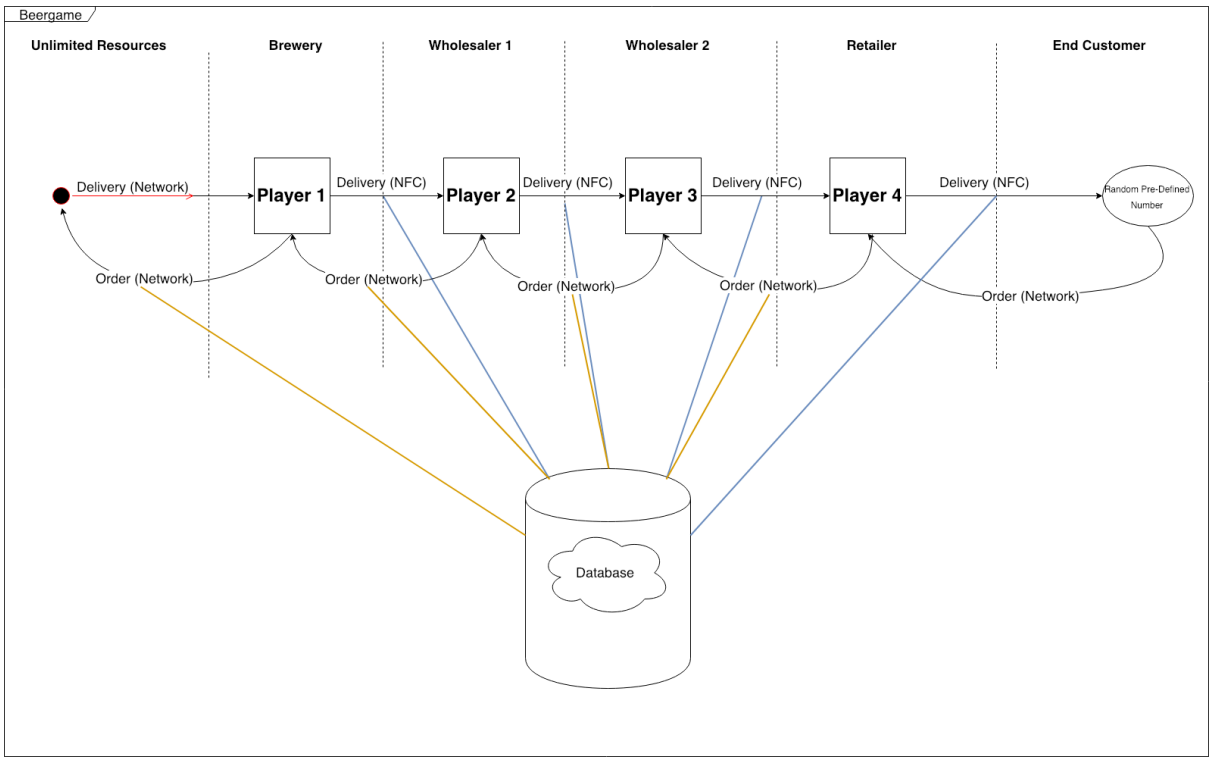

Fig.3: General process

Game Round

Each game round follows the same scheme.
In the beginning of the round, the player receives the order that is placed to him from the next player and the delivery, that he ordered in the previous round.

In order to receive the order, the player is not required to take any further action, his device will automatically receive it via the network and update the respective warehouses accordingly. In order to receive the delivery, he is supposed to place the NFC tag onto the reader of his device and initiate the delivery with a button click. After that, he is ready to take action in the round.

With a glance onto the Open Orders indicators on the screen, he will be able to decide whether there are enough products in the outbound warehouse to fulfill the order or not. If yes, he can directly enter the amount of goods to be delivered and initiate the delivery. If not, he will have to see how many products there are in the inbound warehouse. Depending on that state, he can start production to reach the required amount. After the production has succeeded, he continue as planned with the entering of delivery amounts and finally delivering them. In case there are no goods in the inbound warehouse (and outbound warehouse), the production **and** the delivery becomes impossible and the player must proceed without fulfilling his order.

In the end, the player needs to place an order to the previous player in order to restock his warehouses and stay capable of acting.

The illustration below shows the respective diagram for the described process.

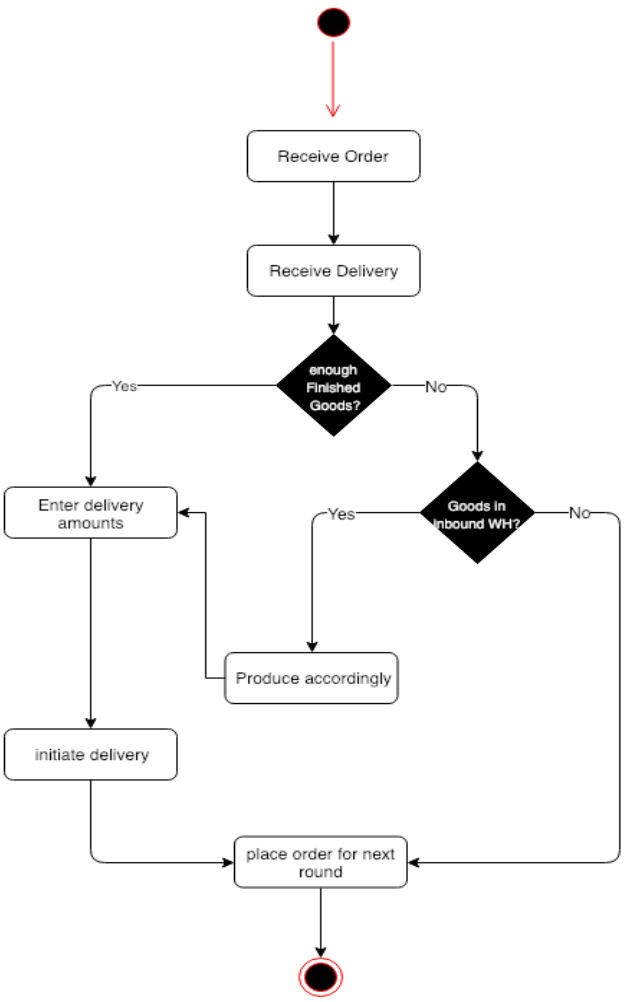

Fig.4: Flow chart

4.1 Database Structure

Below you can see the structure of the relational database, that we have created in order to enable the gameplay:

Master tables:

Team:

Team_id	already taken	Budget	Cost_inbound_warehouse	Cost_outbound_warehouse	Cost-backlog

Products:

Team_id	Product_name	Production_time	Initial_inbound_amount

Rounds:

Number_of_rounds	Round_time

Fact tables:

Orders: (only one order per round and product)

Order_id	of_team	(from_team)	round_number	Product_name	amount	Amount_delivered	closed	timestamp

Deliveries/Outbound: (only one order per round and product)

Delivery_id	From_team	(To_team)	Round_number	Product_name	amount	timestamp

Production: (only one order per round and product)

Production_id	Team_id	Product_name	Round_number	amount	(total production time)	timestamp

Inbound: (what was on the NFC tag)

Inbound_id	To_team	From_team	Product_name	Round_number	Amount	timestamp

4.2 Admin – Config Page

This page provides opportunity for the Admin to provide the teams with their initial values

a. Inventories per Product / Team
b. Production Time per Product / Team
c. Budget per Team
d. Duration per Round (in Seconds)
e. Number of Rounds to be played
f. Cost for each product in the backlog
g. Cost for each product in the Inbound-Warehouse
h. Cost for each product in the Outbound-Warehouse

The Start Button

i. Writes the inserted values into the database
j. Distributes the initial values per team onto the RaspberryPi of the respective Player (Team 1 / 2 / 3 / 4)

After the Start Button has been clicked, the admin **does not have further possibilities to edit the values.** If something has been entered falsely, the game has to be aborted.

4.2.1 Limitations

1. The input fields should only be filled with whole numbers of realistic height
2. Until now, the input fields accept inputs of all data types (Strings, Ints, Doubles etc), which are not necessarily processable of the game's logic. Therefore in further development steps (either when entering the inputs, or when clicking the start buttons) the input fields should check whether correct processable numbers are entered and if not produce an error message.

4.3 Admin – Game Page

This page has three single functions and is to be used by the game administrator

a. Pause (and continue) the Game
b. Abort the Game

4.3.1 Limitations

The Pausing function only stops the timer, the players in this time are required to "play by the rules" and not take any actions when the game is officially paused – **the functions and buttons of the Players' gamepage remain active**. A next step could be to broadcast a message window to the player raspberries with a message "The Game is paused. Please wait until the Admin continues."

4.4 Player – Login Page

This page provides the opportunity for each player to select a team. If a player selects a team and clicks "Login", the database receives this team to be taken. If after that on other player tries to select the same team, he receives an error-message and is advised to select an other team.

4.4.1 Limitations

Until now, the game is limited to max. four players. For the future, one could extend the game's logic to provide the opportunity to play with X players.

4.5 Player – Game Page

This is the page on which the players actually play the game

a. It provides operational information on
    i. how much inventory per product there is in the Raw-Material Warehouse (=Inbound-Warehouse)
    ii. how much inventory per product there is in the Finished-Goods Warehouse (=Outbound-Warehouse)
    iii. How much open deliveries there are in the current round
    iv. How much accumulated open deliveries there are in total (backlog)
b. It provides the possibility to
    i. Enter amounts to be ordered for the next round
    ii. Enter amounts to be produced in the current round
    iii. Enter amounts to be delivered for the next team
c. It provides administrative information on
    i. The current round number
    ii. The current state of the timer (how much time left in the current round)
    iii. Current budget (Initial Budget minus cost of warehouses)
    iv. Production Times per product
d. It provides the possibility to end the round before the timer has counted down completely by clicking on "Done"

4.5.1. Limitations

The input fields should only be filled with whole numbers of realistic height
Until now, the input fields accept inputs of all data types (Strings, Ints, Doubles etc), which are not necessarily processable of the game's logic. Therefore in further development steps (either when entering the inputs, or when clicking the start buttons) the input fields should check whether correct processable numbers are entered and if not produce an error message.
Additionally when clicking "deliver", the program compares the accumulated time the production would need (each unit to be produced multiplied with production time of product) and the remaining time in this round. If the production time exceeds the remaining time, an error message should be displayed.

This function already is implemented, **but if the production time is ok, there is no other time for production started. The production is happening immediately and does not take the time it should need.**

4.6. Player – WaitForEndOfRound Page

This page is being displayed, if a player finishes the round before the official time limit has expired. It disappears automatically as soon as one player clicks "Done" on the Game page (only for this player)

4.7 Player – EndOfRound Page

This page is being displayed for a defined time period at the end of each round in order to give every player the possibility to resettle. After the time has expired, the new round commences.

4.7.1 Limitations

Further development could implement an overview of each round's results on this page, that shows

a. How the own status looks like
b. An intermediate "ranking" of the players based on their remaining budget.

5. Business Intelligence Dashboards

In addition to the information and numbers displayed on the Game Page, a Business Intelligence (BI) system can be featured to enhance the performance and strategy management throughout the game by providing insights and KPI related to the current game. Separate computer devices like laptops are needed to display the individual dashboards for each team. It can be used as an additional tool by the players to include a passive supporting system for the game.
The BI software used for the implementation was Tableau Desktop, a BI and Analytics tool globally utilized by organizations. As described in the "General Process" and "Database Structure" sections, all transaction data are stored centrally in a database on the Admin RaspberryPi device. The database structure was already build in a way that would enable a simple relational data storage and retrieval.
The system running the Tableau software must be connected to the common network of RaspberryPi devices, too, for instance via a local WiFi connection. When setting up the data source connection in Tableau, it is possible to target the central database via the IP address of the Admin RaspberryPi device. The overall setup is depicted below.

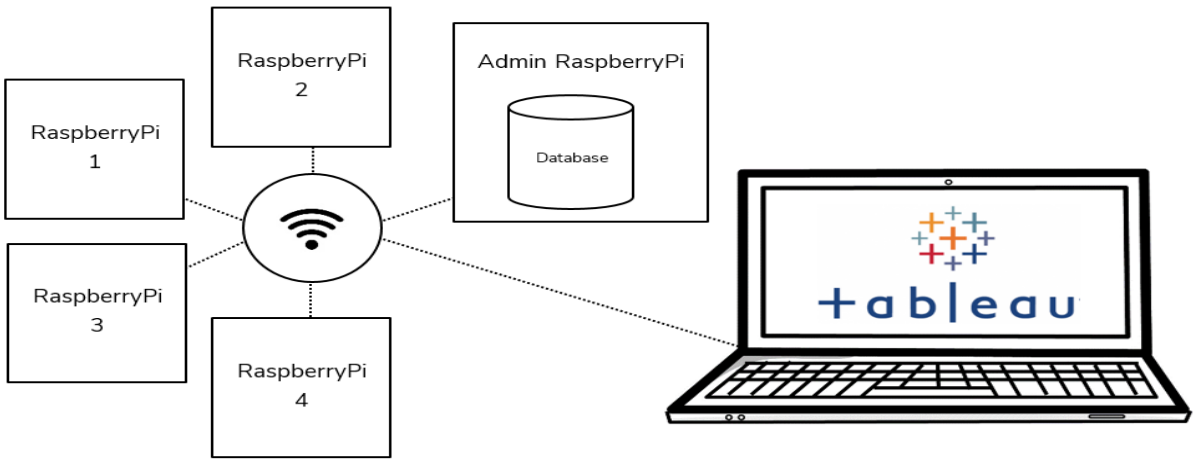

Fig.5: Tableau software used for analysis and visualisation

Using the implemented Tableau dashboard files, the data fields will correspond to the dimensions and metrics used in the visualizations. Therefore, the graphs and figures will dynamically adjust according to the current game's input data in real-time.
Regarding the usage of the Business Intelligence solution during the game, the dashboards differ depending on the user type: player or administrator. Players are supposed to see only their own performance insights, while admins shall be able to monitor and compare the performance of all participating players or teams. Implicitly, the admin BI dashboards contain more information than each player's dashboards. The following dashboards were considered insightful for the users and implemented:

6. Description of Dashboards

6.1. Team Dashboards

Inbound, Outbound and Order Comparison:
The first team dashboard is the 'Inbound, Outbound and Order Comparison'. It shows an overview of the respective amount of inbound products (what is delivered TO the team), outbound products (what is delivered FROM the team) and ordered products (orders created from the team) per round and team. It is recommended to filter the dashboard on products in order for the team to see how each product performed. It gives the teams an overview if their strategy of orders is in line with what gets actually ordered and what they deliver. For the game, the dashboards will be pre-filtered on one team so each team can only see their respective numbers. Only the Admin will be able to see and filter on all teams.

The visualization of the dashboard is three simple bar charts. On the x-axis at the bottom the round numbers are displayed. The x-axis at the top shows the different teams (for the game teams, there will only be one team). The y-axis shows the amount of inbound, outbound and orders.
The dashboard is filterable on products. It is also filterable on teams, but this function will only The KPI's shown in this dashboard are amount of products inbound, outbound and ordered.

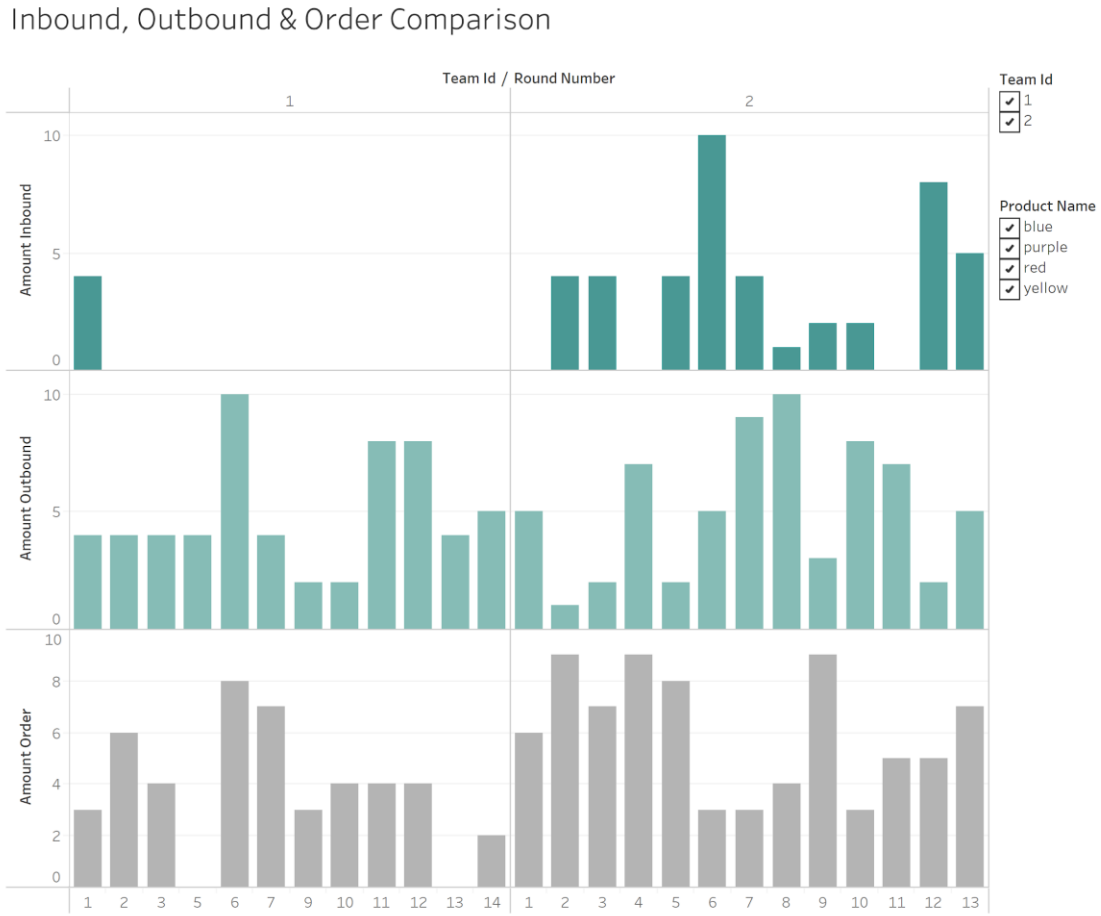

Fig.6: Dashboards showing compaison between inbound, outbound and order placed

Orders received vs. Production:

The second team dashboard shows the 'Orders received vs. Production'. With this dashboard, the teams can compare the numbers of products they produced to the number of orders they received from the next team in the supply chain. It shows the production and orders per product/color and round so the teams can review and analyze their performance at all time. They can see immediately if their production is not in line with the received orders.

The dashboard shows three different visualizations. The main visualizations are two line charts. The upper line chart shows the amount of products produced for each product/color. The second line chart shows the number of received orders for each product/color. In both visualizations, the x-axis shows the round number and the y-axis shows the number of products. The third visualization on the right is a pie chart which shows the total production time proportion. It shows which products needed which amount of time in the production process. The dashboard is filterable only on team, but this function will only be available for the Admin. be available for the Admin who leads the game.

The KPI's used in this dashboard are the amount of products produced and the number of orders received. Furthermore, the total production time for each product is displayed.

Fig.7: Dashboard showing analysis between order received vs products produced i.e production

Orders created vs. Production:

The third dashboard shows the 'Orders created vs. Production'. This dashboard allows the teams to compare their production against the orders they created. It shows the number of orders created and products produced, and it shows the production time of each product used per round. It helps the team to be keep track on their production time per product and round. Furthermore, the teams can review if their orders created are still in line with their production.

The dashboard shows two visualizations. The upper part is a stacked bar chart which shows the number of orders the team created per product and round. On the x-axis the number of products is displayed, the y-axis shows the round numbers. Each color in the bar charts stands for a different product. The second chart is also a stacked bar chart. The x-axis shows the total production time and the y-axis shows the round numbers. The visualization shows how much time of one game round was needed to produce a specific product and by hovering over the chart, more details of the product (e.g. number of products produced) are displayed. The dashboard is filterable only on team, but this function will only be available for the Admin. The KPI's used in this dashboard are the amount of products produced and ordered and also the total production time for each product/color.

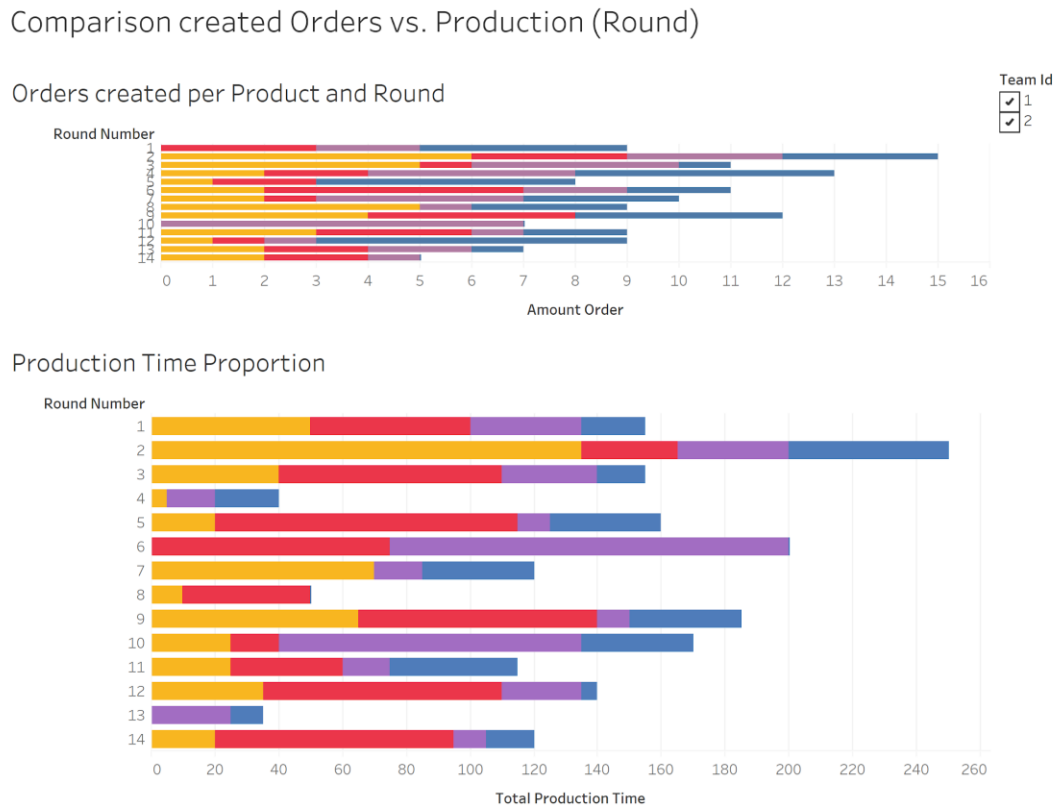

Fig.8: Dashboards showing comparison between created orders vs production (round)

6.2 Admin Dashboards

Production per Team and Round:

The first overview dashboard for the Admin is the 'Production per Team and Round' dashboard. It contains the number of products produced per round and team. For each round, it shows how many products from each color a team has produced. With this dashboard, the Admin can compare the production of the teams throughout the whole game and sees the development of the production. The Admin sees at an early stage if the production of the teams corresponds or if there will be gaps in the delivery at a later stage. He/she also sees when there is an issue during the game rounds, e.g. a team did not order produce anything, then there will be a gap in the dashboard.

The visualization is a stacked bar chart. On the x-axis at the bottom, the different teams are displayed and at the top, the round numbers. The y-axis shows the amount of products which were produced per team and round. The content of the dashboard are stacked bar charts which show all 4 different products/colors. The small visualization on the right is a pie chart which shows the total amount of orders per team. This visualization can be used as a filter to focus the whole dashboard on one specific team.The dashboard is filterable by team, round number and product name so the Admin can have an overview and at the same time is able to focus on

specific dimensions.The KPI's shown in this dashboard are the amount of products produced and the total amount of orders.

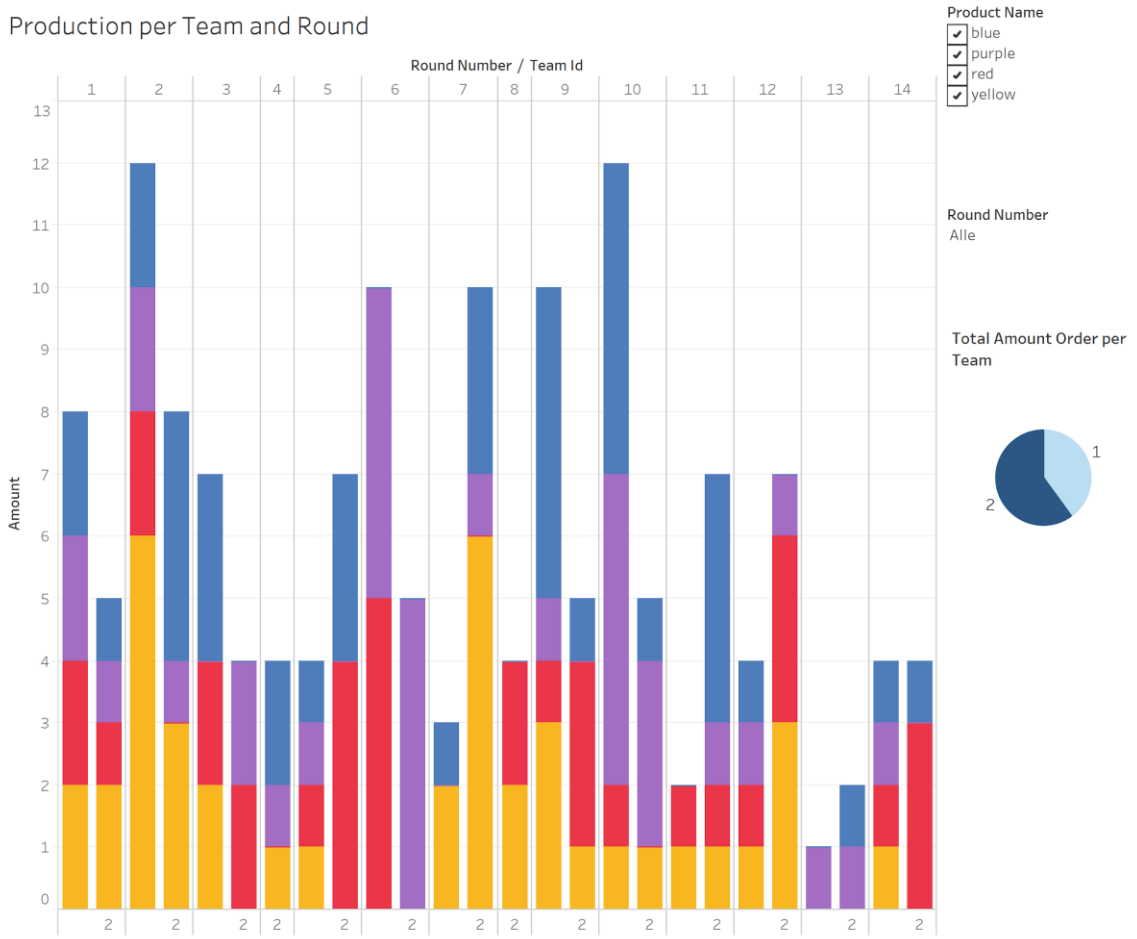

Fig.8: Dashboards showing production per team and round

Team Dashboards for the Admin:

The team dashboards described at the beginning of this chapter can also be used by the Admin. Altogether, they give a good overview of the game and consider different views and dimension. For the Admin the dashboards are not pre-filtered on any team, so the he/she is able to see all teams at once or focus on one specific team.

<u>GeneralInformation</u>
For the dashboards, different worksheets in Tableau are used. There are in total 9 worksheets showing the following information:

- Inbound-Outbound-Order Comparison per Team and Round
- Inbound-Order Comparison per Team and Round
- Total Amount of Order per Team
- Total Production Time Proportion
- Orders created per Product and Round

- Production Time Proportion per Product and Round
- Production per Product and Round
- Orders received per Round and Product
- Production per Team and Round

The worksheets and the dashboards are saved as a Tableau file and can be found in the attachment. The dashboards are developed putting different worksheets together. All dashboards are designed dynamically which means that most of the visualizations can be used as a filter for the overall dashboard. By hovering over the visualizations/charts more detailed information of the specific data points are displayed. For a good understanding of the dashboards it is necessary for the user to get into the details of the beer game itself and how it was developed during this course. For example, the development included possibilities of delay in ordering or producing products. These circumstances have to be taken into consideration when doing the analyses with the dashboards.

7. Other limitations

- Dashboards

The dashboards allow the game players to get a first overview of the data. However, there are multiple other ways to display the data and multiple other possible dashboards. For the beginning, only some first high-level dashboards were developed. More complex views, e.g. a calculation of the total amount of stock in the warehouses (raw materials and finished goods), are out of scope at this point of time. The design of the dashboards also shows room for improvement; only basic features were in scope and the focus was rather on the function of the dashboards than on their design.

- Data Generation and Interpretation

The beer game is originally designed to be played by four teams for its completeness and to observe the effect of supply chain challenges. But we were only able to play with two teams since we had only two working Raspberry Pi's available, so we couldn't generate enough sufficient data to perform meaningful analysis and interpret the results based on the beer game design.

- Historical Analysis

Historical data comparison was also not possible as restarting of the game clears the initially stored data in the database.

- Copying Data

Copying the data to utilise it locally for further analysis offline was only possible by extracting the data source. That means for the user can analyze the data offline, however, the dashboards and the underlying sources cannot be changed offline. To access the live data of the dashboards, the user has to be connected to the database by being connected to the network and Raspberry Pi's. The user needs to have a running version of Tableau to open the dashboard as Tableau file.

- Improvements

More teams can play the game so that sufficient data is captured and better analysis is performed, interpreting better results.
Other Analytical tools like KNIME, Power BI, Click view etc can be used to improve the analytical results.

8. Technical attachment

Documentation of the current configuration of the Raspberry Pi and the network environment

- WiFi-Settings of the Access Point / WiFi Router:
  - Router Login-Password: "HNU2018!"
  - WiFi Name: BigDataAnalyticsLab
  - WiFi Password: HNU2018!
- "Admin" Raspberry Pi
  - Username: pi
  - Password: raspberry
  - Mysql-server is installed
  - Hostname: rp-admin
  - IP: DHCP
- MySQL-Server (which is installed on the "Admin" Raspberry Pi)
  - Username: pi
  - Password: pi
  - Root-account: username: root password: root
  - Config file for the database: /etc/mysql/mariadb.conf.d/50-server.cnf
- Team1-Raspberry Pi and all the other Raspberry Pi's in the game
  - Login: pi - Password: raspberry
  - Hostname: rp-teamX (X is the number of the raspberry, e.g. "rp-team1")
  - IP: DHCP

How the Raspberry Pi's were configured (All necessary steps in case there is an additional Raspberry Pi which has to be installed)

Change the hostname of the Raspberry Pi:

1. Open console (LXTerminal)
2. Sudo nano /etc/hostname → change raspberry to rp-team1 or whatever you want
3. Sudo nano /etc/hosts → change raspberry to rp-team1 or whatever you want
4. Sudo reboot

Install the mysql-database-driver on the Raspberry Pi:

1. Open console (LXTerminal)

2. Sudo apt-get install python-mysqldb
3. Sudo python -m pip install mysql-connector

Install the SQL-Server on the "Admin" Raspberry Pi:

1. Open console (LXTerminal)
2. Install the mysql-server
    a. sudo apt-get install mysql-server
3. Create database user pi:
    a. Sudo mysql -u root -p (enter password 'root')
    b. Create user 'pi'@'%' identified by 'pi';
    c. Grant all privileges on *.* to 'pi'@'%' with grant option;
    d. Flush privileges;
    e. Exit
4. Allow access via Network:
    a. Sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
    b. Change the Bind-address to 0.0.0.0

How to attach and install the NFC card reader to the Raspberry Pi

Wiring of the NFC card reader:

See the instructions of the PDF "RFID_RC522_Manual.pdf" on page 6 from http://anleitung.joy-it.net/wp-content/uploads/2016/09/RFID_RC522_Manual.pdf

Installation of the required Software:

See the instructions of the PDF "RFID_RC522_Manual.pdf" on page 8 from http://anleitung.joy-it.net/wp-content/uploads/2016/09/RFID_RC522_Manual.pdf

How to find out the IP Address of the Raspberry Pi to be able to connect to the Raspberry Pi from your PC after joining the WiFi

1. Open Console (LXTerminal)
2. Type in ifconfig and press enter
3. In the section wlan0 you will find in the second line "inet" something like 192.168.1.33. This is the IP-Address.
4. Use this IP-Address to connect to the Raspberry
5. Username and password: pi