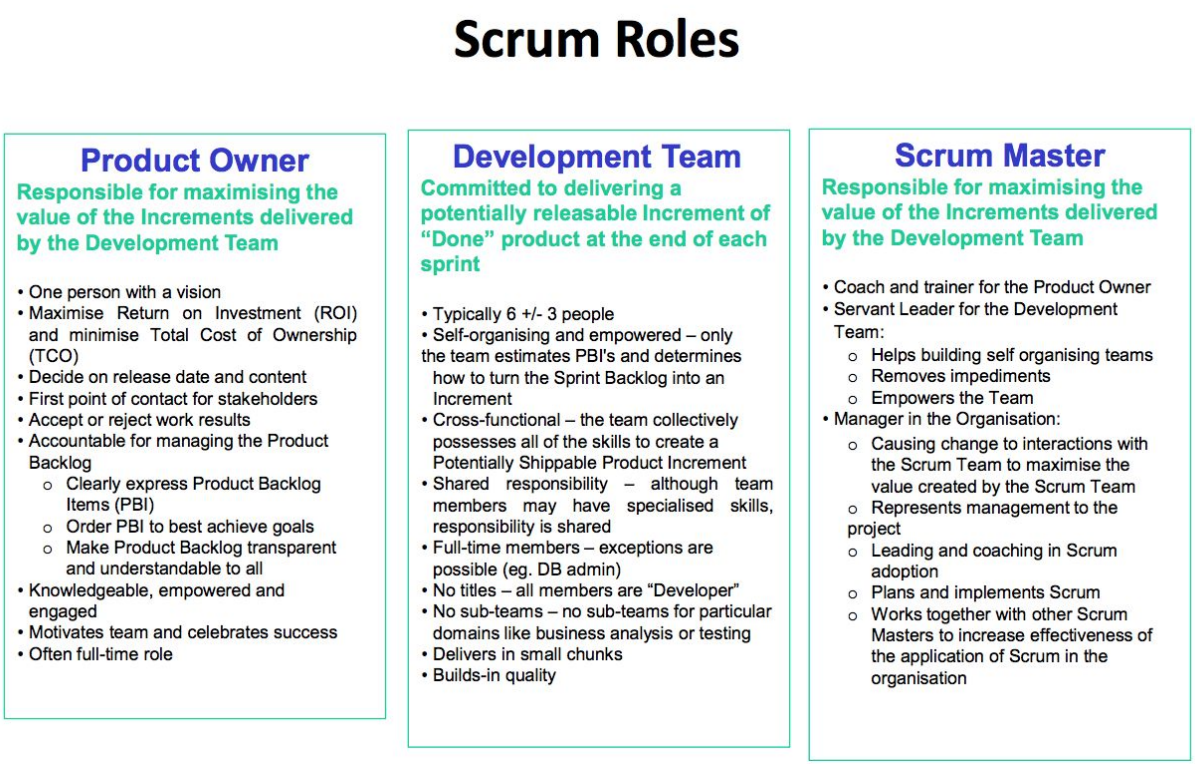
Scrum Roles:
Client/Product Owner: Khrystel
Developers: Kelvin & Aliya
Scrum Master: Zac



Requirements (Database) KS:

1) Record of history of rental services of CRC's customers
    a) Customer information
    b) Car information
    c) Store information
    d) Time
2) Browse the numbers of cars that are picked up or returned in stores monthly
3) Browse car recommendations for customers (based on city information and time)
4) Browse which cars are currently rented out or cars that are available (based on customers, city, and time)
5) Provide information and ALERTS about the vehicle, mainly:
    a) Rego expiration
    b) When service is due
    c) Kilometres
    d) Which parts have been recently renewed or replaced, and dates
6) Petrol cost calculator for customers - suggests which car is cheaper to use based on the travel and fuel

Requirements (frontend web interface) KS:

1) Functional displays such as:
    a) Analysis results
    b) Report functions
2) Access control system
    a) Login with username and password for company users
3) Customer web interface
    a) Lets customers view all the cars that are available to rent (including car information and current location via database connection)
    b) Can reserve a car

User Stories (database) KS:

1) As a product user, I want to be able to record the history of rental services of CRC customers so that I can look back at information when needed.
2) As a product user, I want to able to browse the number of cars that are picked up or returned in stores monthly so that I can determine which cars are used the most.
3) As a CRC customer, I want to be able to browse car recommendations so that I know which car is most suitable for my travel.
4) As a CRC customer, I want to be able to browse which cars are currently available or unavailable so that I know which cars I can rent out.
5) As a product user, I want to be able to get alerts if vehicles are due for servicing, or if registration is expired, so that I am timely efficient.
6) As a CRC customer, I want to be able to calculate which car is the cheapest to travel in based on fuel so that I can save money.

User Stories (frontend web interface) AR:

1) As a developer, I will want to implement test cases throughout the program so that I will be able to write correct working code.
2) As a developer, I want to generate and insert documentation throughout the code so that myself and future developers are able to understand the program's architecture.
3) As a developer, I want to ensure that the interface is user friendly so that customers and employees are not confused when operating it.
4) As a developer, I want to add a functional button that displays analysis results so that those interested in such results can acquire them effortlessly.
5) As a developer, I want to implement a login system so that the program can distinguish users in order to make sure the company's data is being accessed by trusted and known parties.
6) As a developer, I want to develop an interface that allows customers to view all available cars so that the customer can see what car they can hire.

Appendices

Criteria for User Stories (HD):

- Above 85% of your stories demonstrate very good application of all the principles of INVEST and each story is a clear expression of a single idea behind the requirement.
- Above 85% of your stories represent a wide range of features that have a balance of moderate to challenging requirements that deliver high business value.
- All stories have been prioritised using MoSCoW as clearly indicated on card.
- All stories have been realistically and consistently estimated with story points.
- Above 85% of stories have acceptance criteria that provide a clear understanding of the client's goal & a clear boundary of scope.
- Above 85% of acceptance criteria can clearly be implemented as tests.

What is INVEST?

I = Independent __ Allows a story to he worked on independently without affecting the others. Avoid any kind of ordering or number system... Ensures that one story does not hold up the completion of the others.

N = Negotiable __ Don't make a story a ironclad set of features... Leave some room to alter later on, just enough to explain the underlying concept.

V = Valuable __ Ensure that all stories bring value to a client. A homepage without definition has little value... For instance expand acceptance criteria to include summaries of deals or promotions, etc

E = Estimable __ Don't make a story too vague, a story needs an estimated workload of some kind. This and small are often influenced by one another.

S = Small __ Don't make a story too broad... They need to be specific... For instance an Intuitive GUI influences all stories and it's not independent either. Where as one tailored towards accounts tends to be independent based on acceptance criteria.

T = Testable __ A story that contains terms like Intuitive and Simplistic need to be avoided as your definition could be different to the client. Ensure the acceptance criteria can be tested... By all means have those terms in the the title but not the body or AC.

What is MoSCoW?

Must have – Should have – Could have – Won't have, prioritising what needs to be done