**Nulscracy Roles and Rewards**

**1] Introduction**

Current reward system is not scalable nor fair, at some point we will need to change the terms for current and future CCC members which is something we can and must avoid at all cost, mainly because in order to attract talented people we need to offer a clear and sustainable reward system, so they can project their investments in money and time medium and long-term.

The purpose of this proposal is to establish a scalable and fairer model based on Roles that is allow to expand and contract seamlessly in shorts periods of time.

Each Role will have attached a reward scale that will be adjusted by several parameters described in section 4].

**2] Role description**

Each Role has several fields that can be filled in a table format. The example below will be used to explain each field.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **02.01-01** | | | | | | | | |
| **Name** | **CCCd Lead Link** | | | | | | | | |
| **Parent Circle** | **01 Circle’s Council** | | | | | | | | |
| **Period** | **3 Months** | | | | | | | | |
| **Assignment Type** | **Voluntarily** | | **Shift** | | **Random** | | **Assigned** | | |
| **Activities** | **02.01-001** | Communicate properly all decisions and policies from Circle’s Council to other Circles (including Community Council) and vice versa | | | | | | | |
| **02.01-002** | Report progress from each member to Link Lead of Community Circle’s Council. | | | | | | | |
| **Requisites** | **Commitment** | Fulltime | | Part time | | Per Task Basis | | | |
| **Sub Circles** | 02.01 - Council Member | | | | | | | |
| 02.02 – Management | | | | | | | |
| **Roles** | - | | | | | | | |
| **Skills** | Needs to have knowledge about all internal Roles, basic knowledge about all Circles | | | | | | | |
| **Levels** | **01 - Basic** | At least 3 months working as CCCd member | | | | | | **PRS** | 0 |
| **SRS** | 5000 |
| **PTB** | **-** |
| **02 - Intermediate** | At least 1 year working as CCCd member | | | | | | **PRS** | 0 |
| **SRS** | 10000 |
| **PTB** | **-** |
| **03 - Advanced** | At least 3 years working as CCCd member | | | | | | **PRS** | 0 |
| **SRS** | 15000 |
| **PTB** | **-** |
| **04 - Expert** | At least 5 years working as CCCd member | | | | | | **PRS** | 0 |
| **SRS** | 20000 |
| **PTB** | **-** |

2.1] ID

This is a string composed with two parts that identifies a Role without ambiguity. The first part is called ‘*scope*’ and is crafted using the concatenation of the circles on which the Role belongs, in the example above 02 is the code of the Green Circle (CCCd) and 01 is the code of the Circle’s Council. The second part is just a sequential number that increments by one each time a new role is created in the same *scope*.

2.2] Name

The name of the Role needs to be unique among all Circles, it should describe in as few words as possible the main objective of the Role.

2.3] Parent Circle

The code of the Circle on which this Role belongs. In the example the Lead Link Role belongs to the Circle’s Council which has code 01.

2.4] Period

Represents the minimum period of time on which the member should fulfill the Role’s responsibilities.

2.5] Assignment Type

Defines the different ways a member can take this Role, there are four methods:

2.5.1] Voluntarily: This is the preferred method and happens when someone takes this Role on voluntary basis.

2.5.2] Shifts: If more than one member fulfills the requisites then it is possible to take shifts between periods.

2.5.3] Random: Another method is to choose on random basis.

2.5.4] Assigned: When some entity or person forcibly assigns this Role to a member, usually this type should be only used when the Role is not taken using the other methods.

In the example all four methods are possible and are listed from the most preferable to the least preferable.

2.6] Activities

It’s a list of responsibilities that a member must fulfill, if there are too many responsibilities then it would probably be a good idea to split the Role in two.

2.7] Requisites

There are four types of requisites that a member may need to fulfill to be a valid candidate for a specific Role.

2.7.1] Commitment: There are three types of commitments, those who work full time, those who work part time and finally "per task basis" that are ideal for bounties. Translator Role could be a good example of a "Per Task Basis" type of commitment.

In the example only full time and part time members are allowed to take this Role.

2.7.2] Sub Circles: This parameter specifies on which Sub Circles a person needs to be a member of to be a valid candidate. For Lead Link Role it is necessary that a candidate must be a member of Circle’s Council and also of the Management Sub Circle.

2.7.3] Role: If Sub Circles are not defined, it is possible to define specific Roles as requisites for candidates.

2.7.4] Skills: An optional list of skills that a candidate must have to adequately fulfill the Role.

2.8] Levels

Levels represent the expertise level that a member can have, the more expertise the more the reward. Each Role must define at least one level. Each level has its own reward scale that it has to be multiplied with the FBI calculated at the beginning of the month. Rewards scale has 3 types:

2.8.1] Primary Reward Scale (PRS): New Members need to choose one base Role to establish its basic reward level. Primary Reward represents the main basis of income for the respective Member. If a Role does not have a Primary Reward Scale, it means it can’t be chosen as a base Role. The number represents the amount of NULS needed to be staked on the members agent.

2.8.2] Secondary Reward Scale (SRS): When Members want to perform other Roles besides the basic one, their extra reward Levels are established by this parameter. Members need approval from Circle’s Council to perform other Roles besides his basic one.

2.8.3] Per Task Basis (PTB): This type of rewards is chosen when the task does not require continuous work but only specific tasks, for example translating the whitepaper would suit well in this type of reward. The reward is fixed and payed only once, it can also be payed in several periods depending on the exact nature of the task.

*What is missing?*

Two fields are missing on purpose from the example because it would add unnecessary complexity at this instance, those will be described briefly below so they can be implemented on further revisions of the model.

2.9] Metrics

Every activity on a Role should have at least one metric assigned so it would possible to collect important information about time and performance, this data will be invaluable in the future so we can adjust policies based on statistical models and not only on empty assumptions.

2.10] Audit Parameters

Using metrics described above, in the future it will be possible to define a list of parameters for each activity that an Auditor can evaluate consistently and efficiently the performance of each member for each activity

**3] Role example 2**

**We will work out on of the most important Roles in CCC which is the Developer Role**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **02.02.01-01** | | | | | | | | |
| **Name** | **CCCd Developer** | | | | | | | | |
| **Parent Circle** | **01 Development** | | | | | | | | |
| **Period** | Indefinite | | | | | | | | |
| **Assignment Type** | **Voluntarily** | | **Shift** | | **Random** | | **Assigned** | | |
| **Activities** | **02.02.01-001** | Translate the design made by the Software Architect to a development language | | | | | | | |
| **02.02.01-002** | Report progress to the Project Manager | | | | | | | |
| **02.02.01-003** | Support and maintain the final product | | | | | | | |
| **Requisites** | **Commitment** | Fulltime | | Part time | | Per Task Basis | | | |
| **Sub Circles** | - | | | | | | | |
| **Roles** | - | | | | | | | |
| **Skills** | Needs to have deep knowledge on a specific development language. A lot of experience developing quality code. | | | | | | | |
| **Levels** | **01 - Basic** | Proven experience implementing high quality code | | | | | | **PRS** | 220000 |
| **SRS** | 10000 |
| **PTB** | **-** |
| **02 - Intermediate** | At least 1 year working as CCCd member. Full time member. Basic documentation | | | | | | **PRS** | 250000 |
| **SRS** | 11000 |
| **PTB** | **-** |
| **03 - Advanced** | At least 3 years working as CCCd member. Full time member. Intermediate documentation | | | | | | **PRS** | 300000 |
| **SRS** | 12500 |
| **PTB** | **-** |
| **04 - Expert** | At least 5 years working as CCCd member. Full time member. Full documentation. | | | | | | **PRS** | 350000 |
| **SRS** | 15000 |
| **PTB** | **-** |
| **05 - Master** | At least 8 years working as CCCd member. Full time member. Full documentation. Passing external auditory of the code. | | | | | | **PRS** | **400000** |
| **SRS** | **20000** |
| **PTB** | **-** |

**4] Reward Calculation**

The rewards formula needs to consider an adjustment in case the price is too low in the same way that is calculated now, it also needs to be taken into account that community funds are finite so if the number of developer increases, the total reward output should start decreasing smoothly, there are other factors that needs to be considered explained below grouped in three different parameter sections.

(Before continuing, please open the spreadsheet below for a clearer explanation) <https://docs.google.com/spreadsheets/d/1XYRKXWbrbMBW9fXqapmmuTMntHIoAa3tefbtSPQdOZo/edit#gid=802597527>

4.1] Base parameters: Usually these parameters need to be set just once but can be changed as required

* Initial Developers (IniDev): Defines the minimum number of developers that CCC needs to operate. The default is 10
* Increase 1 dev per ### USD (IncDev): The model proposes a maximum number of developers according to a price range, this parameter states that in order to increase one developer, the price needs to increase ### USD. The default is 0.33 Usd/NULS
* Part Time Multiplier (PTM): Members that work in their free time earns less than members that work full time, this parameter establish the proportion. For example, a value of 0.5 means that part-time members earn just half of what full-time members earn. 0.5 is the default.
* Max Possible Payout For Part Time (USD): Specifies that amount of rewards earned equivalent in USD. The defaults is 7000USD which means that a member that works part time can not earn ore than 7000 USD equivalent in NULS.
* Min Price To Compensate (MPC): The minimum price that the model will compensate if prices are low, the default is 0.6 which is exactly the same as it is now.
* Role Maximum Rewards (RMR): The maximum cumulative staked NULS that a member can earn. For example, if a member takes 4 Roles the maximum number of NULS to be staked is defined by this parameter. The default is 400 000

4.2] Monthly Global Parameters: These parameters need to be revised once per month, but they work for all CCC members.

* Total Community Funds (TCF): The total amount available in the Community Funds wallet including the extra amount staked.
* Current CCC Members (CCM): The current number of members that CCC has.
* 30 Day Simple Moving Average (SMA30): The average NULS price in USD for the past 30 days.
* 200 Day Simple Moving Average (SMA200): The average NULS price in USD for the past 200 days.

4.3] Individual CCC Member Parameters: These parameters need to be revised for each CCC member.

* Total Role Rewards (TRR): This is the cumulative number of NULS that need to be staked for a specific CCC member.
* Full Time? (FT): If the member works full time the value is 1, otherwise it is 0.

4.4] Calculated values: With the parameters above it is now possible to make some calculations

* Max Non Adjusted Staked: This is an intermediary variable needed for further calculations, it represents that theoretical maximum of NULS needed to stake with the current amount of CCC members.
* Suggested Maximum Of Members: The model proposes a maximum limit of members according to the price range, it is important to follow this guideline so the rewards do not dilute too much giving CCC members the ability to project their earnings.
* Fair Base Index (FBI): This is a multiplier that adjusts the price based on the parameters introduced previously.
* Total to be staked (TBS): The TOTAL amount that it is needed to be staked for each CCC member.

**5] Reward Calculation Example**

For the example, lets make the following assumptions:

Warren Buffet finally sees the light and decides to become a CCC developer, he chooses Role CCCd Developer (02.02.01-01), despite his age after a year he is considered an Intermediate Level developer, as this Role is his primary Role, the Primary Reward Scale (PRS) is chosen with a 250000 to be staked. Also, in the coming months he will act as Lead Link for CCCd Circle at intermediate level since he already is developing one year for CCC, as this is a secondary role, the Secondary Reward Scale is chosen (SRS) with 10000 NULS to be staked.

In total, the preliminary reward is the result of staking 250000 + 10000 = 260000 NULS, but this value needs to be adjusted according to current environment conditions so lets also assume the following:

Total Community Funds: 20 000 000

Current CCC Members: 200

SMA30 (USD): 92

SMA200 (USD): 72

**The total Amount to be staked this period for Warren Buffet is 43333 NULS.** The model assures no matter the number of members, this way we can scale even to thousands of members seamlessly.

(You can introduce the values I the link provided)

**6] NULS Suggested Roles**

**6.1] Development sub circle**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **02.02.01.01** | | | | | | | | |
| **Name** | **CCC Software Developer** | | | | | | | | |
| **Parent Circle** | **Development** | | | | | | | | |
| **Period** | Indefinite | | | | | | | | |
| **Assignment Type** | **Voluntarily** | | **Shift** | | **Random** | | **Assigned** | | |
| **Activities** | **02.02.01.01.001** | Translate the design made by the Software Architect to a development language | | | | | | | |
| **02.02.01.01.002** | Report progress to the Project Manager | | | | | | | |
| **02.02.01.01.003** | Support and maintain the final product | | | | | | | |
| **02.02.01.01.004** | Troubleshoot and debug. | | | | | | | |
| **Requisites** | **Commitment** | Fulltime | | Part time | | Per Task Basis | | | |
| **Sub Circles** | - | | | | | | | |
| **Levels** | **01 – NULS Trainee** | Proven experience implementing quality code.  Needs to have deep knowledge on a specific development language. Mastering at least one IDE and tool chain.  Cross platform experience. | | | | | | **PRS**  **SRS**  **PTB** | 250000  10000  **-** |
| **02 – NULS Junior** | At least 1 year working in CCC.  Full time member**.**  Proven ability to debug.  **Role:** Code Documenter Level 1.  **Role:** Tester Level 1.  **Role:** DB Manager level 1. | | | | | | **PRS**  **SRS**  **PTB** | 270000  11000  **-** |
| **03 - NULS Intermediate** | At least 2 years working in CCC.  Proven ability to troubleshoot.  He is familiar to all NULS API methods.  Understands how NULS works internally.  Significant contributions to current product / modules.  Completed at least one NULS module.  Proven ability to break problems into smaller and more manageable tasks. | | | | | | **PRS**  **SRS**  **PTB** | 290000  12000  **-** |
| **04 - NULS Advanced** | At least 3 years working as CCC member.  Completed at least one product.  Experience working with Agile methods.  Mastering of the primary programming language. | | | | | | **PRS**  **SRS**  **PTB** | 310000  **-**  **-** |
| **05 – NULS Proficient** | At least 4 years working as CCC member.  **Role:** Code Documenter Level 2.  **Role:** Dev Ops operator Level 1.  **Role:** Tester level 2.  **Role:** Mentor level 1.  **Role:** DB Manager level 2.  **Role:** Project Manager level 1. | | | | | | **PRS**  **SRS**  **PTB** | 330000  **-**  **-** |
| **06 – NULS Expert** | At least 5 years working in CCC.  Product is used by at least 5000 users or 20 companies.  Product has its own SDK | | | | | | **PRS**  **SRS**  **PTB** | 350000  **-**  **-** |
| **07 – NULS Advanced Expert** | At least 6 years working in CCC.  **Role:** Code Documenter level 3.  **Role:** Architect level 1.  **Role:** Dev Ops operator Level 2.  **Role:** Tester level 3.  **Role:** DB Manager level 3.  **Role:** Mentor level 2. | | | | | | **PRS**  **SRS**  **PTB** | 370000  **-**  **-** |
| **08 – NULS Master** | At least 7 years working in CCC.  Product is used by at least 30000 users or 100 companies.  Product passed external audit of the code by an accredited company.  Product has its own development framework. | | | | | | **PRS**  **SRS**  **PTB** | 390000  **-**  **-** |
| **09 – NULS Grand**  **Master** | At least 8 years working in CCC.  **Role:** Architect level 2.  **Role:** Dev Ops operator Level 3.  **Role:** Dev Ops engineer Level 1.  **Role:** Project Manager Level 2.  **Role:** Mentor level 3. | | | | | | **PRS**  **SRS**  **PTB** | 410000  **-**  **-** |
| **10 – NULS Legend** | At least 10 years working in CCC.  Product is used by at least 100000 users or 1000 companies.  **Role:** Architect level 3.  **Role:** Dev Ops engineer Level 3.  **Role:** Project Manager Level 3. | | | | | | **PRS**  **SRS**  **PTB** | 430000  **-**  **-** |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **02.02.01.02** | | | | | | | | |
| **Name** | **CCC Software Architect** | | | | | | | | |
| **Parent Circle** | **Development** | | | | | | | | |
| **Period** | Indefinite | | | | | | | | |
| **Assignment Type** | **Voluntarily** | | **Shift** | | **Random** | | **Assigned** | | |
| **Activities** | **02.02.01.02.001** | Develop high-level product specifications with attention to system integration and feasibility | | | | | | | |
| **02.02.01.02.002** | Communicate successfully all concepts and guidelines to CCC Council | | | | | | | |
| **02.02.01.02.003** | Define all aspects of development from appropriate technology and workflow to coding standards | | | | | | | |
| **02.02.01.02.004** | Collaborate with other professionals to determine functional and non-functional requirements for new software or applications | | | | | | | |
| **02.02.01.02.005** | Use tools and methodologies to create representations for functions and user interface of desired product | | | | | | | |
| **02.02.01.02.006** | Oversee progress of development team to ensure consistency with initial design | | | | | | | |
| **02.02.01.02.007** | Ensure software meets all requirements of quality, security, modifiability, extensibility etc. | | | | | | | |
| **02.02.01.02.008** | Approve final product before launch | | | | | | | |
| **Requisites** | **Commitment** | Fulltime | | Part time | | Per Task Basis | | | |
| **Sub Circles** | - | | | | | | | |
| **Skills** | * Experience with software development lifecycle (SDLC) processes * Experience with service oriented architectures (SOA). | | | | | | | |
| **Levels** | **01 – Junior Architect** | At least 3 years working in CCC  Full time member**.**  Broad and extensive knowledge of the software development process and its technologies.  Understanding of various coding methods and computer languages.  Data modeling and database design experience.  Knowledge of architectural styles and design patterns.  Completed design of at least one module.  **Role:** Code Documenter Level 2  **Role:** Software Developer Level 3  **Role:** Software Tester Level 2 | | | | | | **PRS**  **SRS**  **PTB** | 300000  20000  **-** |
| **02 – Application**  **Architect** | At least 4 years working in CCC.  Experience with micro services architecture.  Experience with software development lifecycle (SDLC) processes.  Experience with service oriented architectures (SOA).  Completed design, implementation and deployment of at least one application / product.  **Role:** Software Tester Level 3.  **Role:** Code Documenter Level 3  **Role:** DB Manager level 2. | | | | | | **PRS**  **SRS**  **PTB** | 330000  25000  **-** |
| **03 – Solution Architect** | At least 5 years working in CCC.  Completed design, implementation and deployment of at least one complete solution on a company. | | | | | | **PRS**  **SRS**  **PTB** | 360000  **-**  **-** |
| **04 – Enterprise Architect** | At least 7 years working as CCC member.  Completed design, implementation and deployment of at least 5 complete solutions on different companies. | | | | | | **PRS**  **SRS**  **PTB** | 400000  **-**  **-** |

**6.2] Examples**

6.2.1]

On August 2019 **Pablo** was accepted as new CCC member and he chose **Software Developer Role (02.02.01.01**) as his primary Role. As he is beginning his journey in NULS he starts at level **01 – NULS Trainee.**

According to de Software Developer Role card his Base Role Rewards are the rewards generated by 250 000 NULS because that Role is his primary Role we use the PRS value (Primary Reward Scale).

But this is only the Base rewards scale which means that this number should be modified according to several variables presented in the suggested model at:

<https://docs.google.com/spreadsheets/d/1XYRKXWbrbMBW9fXqapmmuTMntHIoAa3tefbtSPQdOZo/edit#gid=802597527>

Each month we must modify the following variables: (also we are filling them with values to complete the example)

Current CCC Members: 15

SMA30 (USD): 1.4

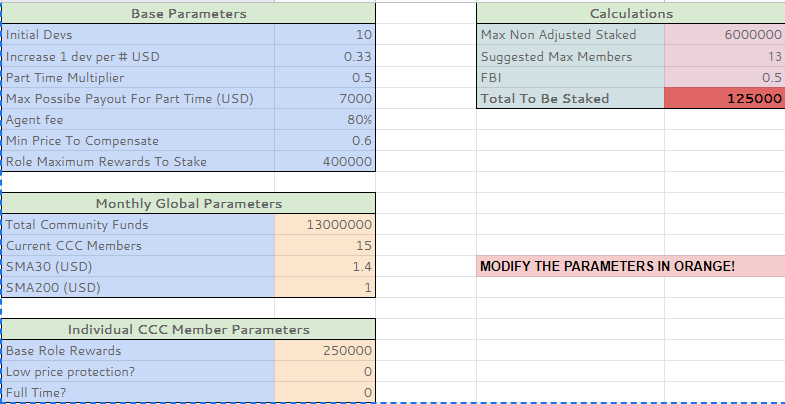
SMA200(USD): 1

Base Role Rewards: 250 000

Low price protection?: 0 (False)

Full Time?: 0 (False)

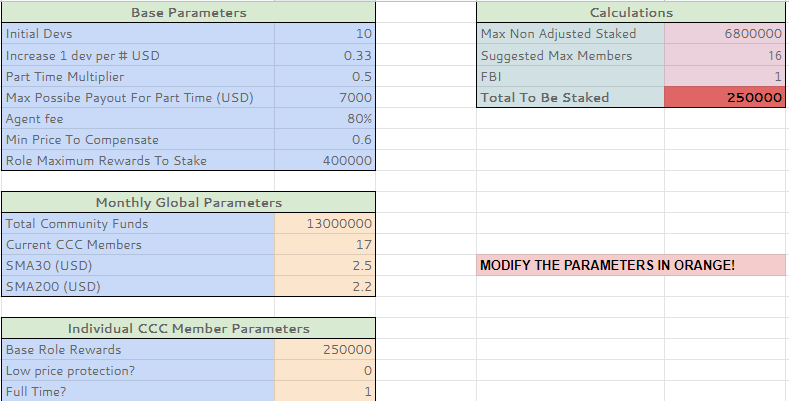
The result according to the model is that Pablo should receive the rewards generated by staking **125000 NULS** with the agent charging 80% fee.



6.2.2]

After a few months Pablo showed that he is very capable so CCC Council asked him if he wants to be a Full Time CCC member, as the price has risen he agreed with this so for the next month his rewards are adjusted again.

In this case we set the variable Full Time? to 1 and now he will receive rewards equivalent to **250 000 NULS** staked in a 80% fee agent:



Notice that in this case the Base Role Rewards is exactly the same as the Total To Be Staked variable, the reason is that as CCC have still few members then its not necessary at this instance to adjust the total rewards because the model assures that the Total Community Funds is enough to cover the stakes of all developers at this instance.

For the next examples we will assume that other Roles are defined entirely, specifically:

Secondary Reward Scales (SRS) of:

**Role:** Code Documenter Level 1 = 10000

**Role:** Tester Level 1 = 10000

**Role:** DB Manager level 1 = 10000

6.1.3]

After a while and a lot of work, Pablo decides that he will properly document his code and after some practice he asks permission to CCC Council to be allowed to perform a Secondary Role, in this case Code Documenter Level 1. CCC Council reviews his performance and also that he is prepared to fulfill the requirements so it accepts his request again!

Now his Base rewards are increased the next month:

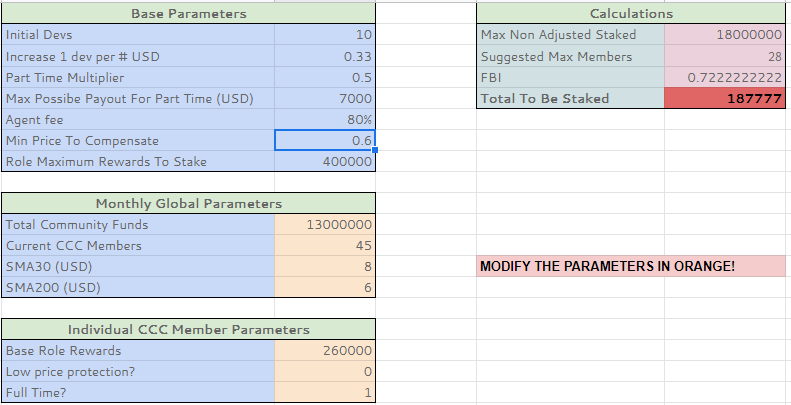
Role: Software Developer Level 1: 250000 (As this is his **primary** Role we use the Primary Reward Scale **PRS**)

+ Role: Code Documenter Level 1: 10000 (As this is his **secondary** Role we use the Secondary Reward Scale **SRS**)

= 260 000 NULS staked with 80% agent fee.

But this is are only the Base rewards! We need to feed the model again with relevant data, for the example we will assume that the number of Current CCC members at that time is **45**

This time the **Total Rewards To Be Staked is just 187 777 NULS!**



What happened this time? Well the model must assure that the amount of NULS available to be staked covers ALL CCC members and this time as CCC members increased to 45 it reduced the total pay for each member proportionally.

This makes us formulate the following question, does CCC members will gain less and less as new members come on board?

The answer is Yes and No at the same time. Its Yes because they will earn less NULS due the amount of NULS to be staked is limited to 13 000 000. On the other hand, they will receive more USD because the price at that point should increase and that’s the main reason **we need to make sure that the value of Suggested Max Members in the model is respected**, this will assure that even earning less NULS each member will earn more USD.

In time this will allow a lot of top talent around the world to join CCC increasing its intrinsic value because the amount of code throughput and quality will increase as well, giving us the ability to create and maintain more projects and ready to use solutions which in turn will make the price raise again, so we will be able to hire more very talented CCC members.

6.1.4]

After more than a year of hard work Pablo believes its ready to raise its level to **Level 2 – NULS Junior,** CCC council reviews that he satisfies the necessary requisites and approves his request (we will assume that Pablo has increased his participation and now he is a Tester Level 1 and a DB Manager 1 as well, so he can satisfy the requirements for Level 2)

For the next month the new Base Rewards for Pablo are the following:

Level 2 Software Developer: 270 000

**Role:** Code Documenter Level 1 = 10000

**Role:** Tester Level 1 = 10000

**Role:** DB Manager level 1 = 10000

Total: 300 000 NULS

We feed the model again with this information and this time Pablo receives the amount of **216 666 NULS** staked on a 80% fee node.

