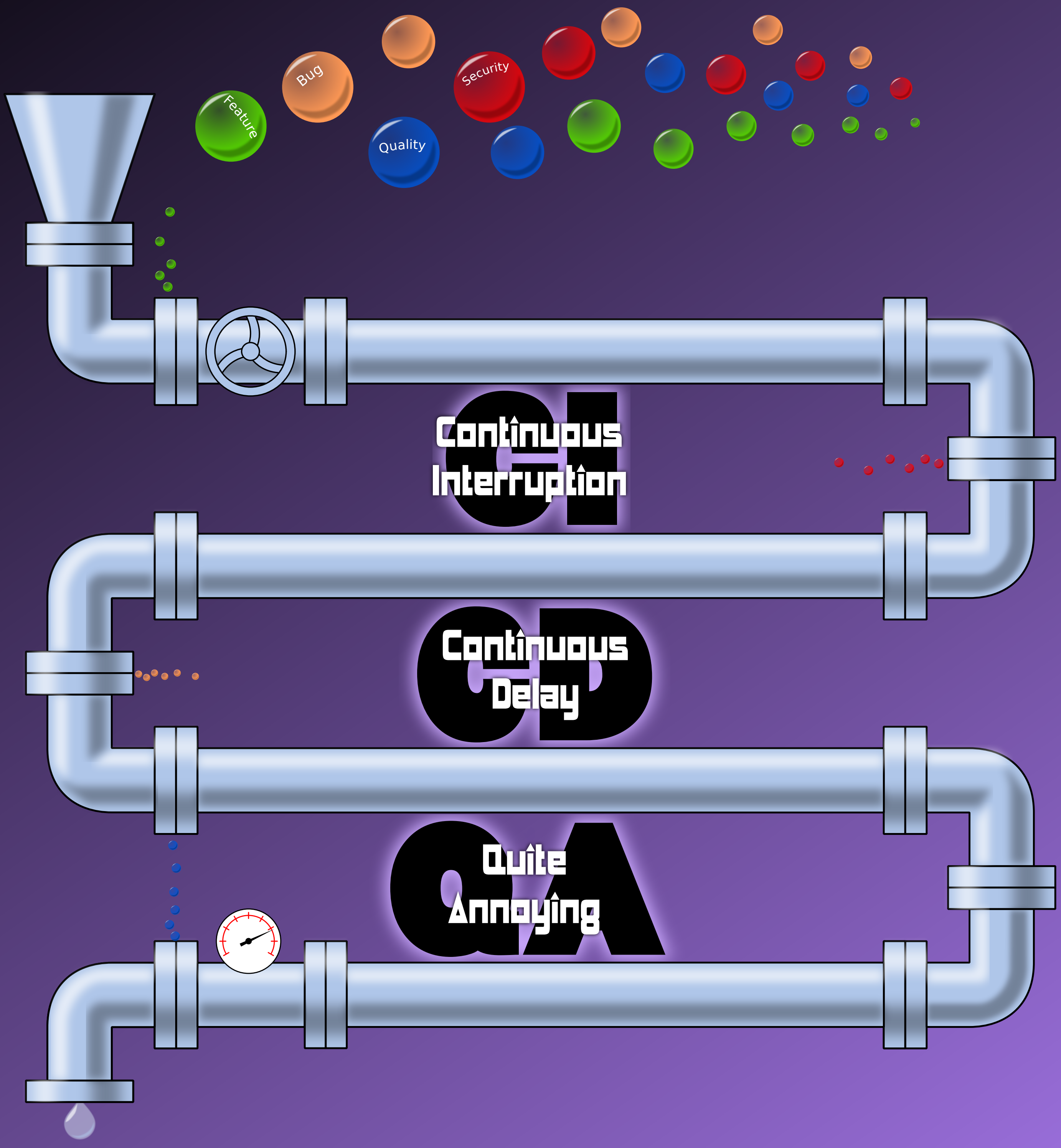
Facilitation guide



SimuLearn

This page is deliberately left blank

Continuous interruption - Continuous Delay - Quite Annoying

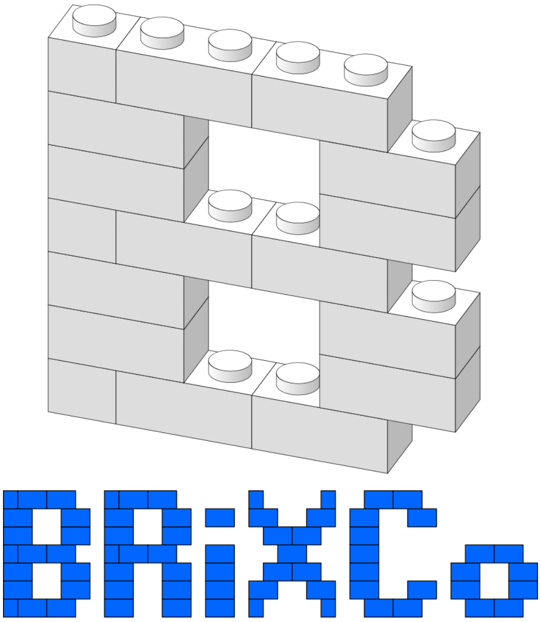
Card game to experience the collaboration between development team, business, CISO and release management

Table of content

Version history

Version	Date	Changes
0.1	2023-05-31	Initial version
0.2	2023-06-24	Adaptations after first test rounds
0.3	2023-08-25	Story telling, personas, processing feedback from another try-out
1.0	2023-09-18	First final version

The story of BrixCo



Welcome to BrixCo, the market leader in toy bricks. We offer the best brick sets in various themes, but also starter kits or even the tiniest individual bricks and the largest base plates you need to realize your own projects. That's why we are BrixCo: we help you build your imagination, brick by brick.

We are going to add a set of groundbreaking features to our online customers platform. We want to help our customers to manage their bricks and sets. And at the same time we will know more about what they currently have, what their interest is, so that we can use this information for marketing purposes. Our customers will be stunned!

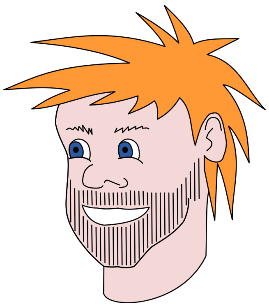
We need to be the first in the market with this solution. But… according to our compliance staff we cannot make concessions on quality nor security! Will you manage to deliver this now groundbreaking product? Or will complying to the security policies and release process cause continuous interruptions and continuous delays?

Time to meet the team…

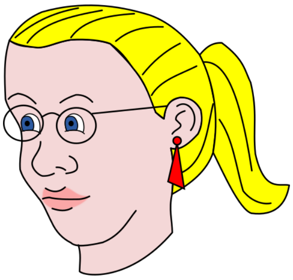
Characters - Round 1

In round 1 the development team will implement the feature requests and fix the reported bugs of the product owner and at the same try to deal with the requirements of both the security officer and the release manager. Let’s see who they are…

Dave - backend developer



This is Dave. Dave is a very skilled backend developer. His code is always well written, covered by a lot of unit tests, and easy to maintain and to extend. When it comes to database performance skills, Dave is the one to rely on.



Debby - full stack developer

This is Debby. Debby is a full stack developer. Like Dave, she is a T-shaped profile. She is not only great at developing both front-end end and back-end. She also has excellent UX skills. When it comes to usability, Debby is the one to call.

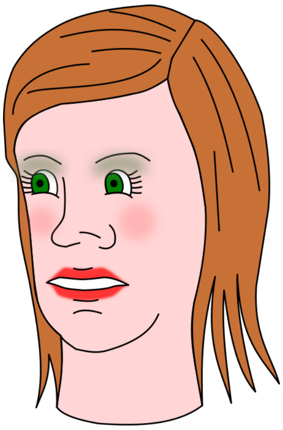
Steve - the Security Officer

This is Steve, the security officer of BrixCo, Also known as Scary Steve.
His favorite expression is: “You must comply with the security policies, or else…!”

> *“Prove that your architecture is able to block threats from outside!*
> *Don’t just implement the happy path, what will someone with malicious intent try to do?*
> *Show me your secure coding guidelines and prove that you apply them!*
> *Make sure the open source libraries you use don’t have security vulnerabilities!*
> *Scan your source code for possible vulnerable constructions*
> *Don’t forget to connect to the central log aggregation service to detect suspicious behavior in your application!*
> *And where are the results of your penetration tests?”*

His interventions tend to **continuously interrupt** the flow of the developers.

Rachel - the release manager

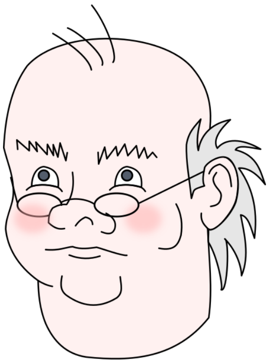


Rachel is the release manager. She does what she had to do: making sure that the developers deliver a quality release increment of the product.

“Have you determined your release date? Where is your release change ticket in the tracking system?
Make sure your dependencies are covered!
Show me your test progress report!
You need to start the Business as Usual tests for this release!
Show me the open defects? Are there still critical bugs?
I want to see the results of your performance tests!
Time to come to the final go/no go meeting!”

Her quality and release related requirements often **cause delays** in the delivery process, because of the administrative overload.

Paul - the product owner



Finally, meet Paul, the product owner of BrixCo. He is a very loyal employee of BrixCo. He has a lot of experience, knows the company’s ins and outs. He is best fit for the role of product owner.

He wants to bring a quality product to the market, with killer features. But things don't go fast enough for him. The requirements of release management and the security officer seriously impact the cycle time of his features, and that’s **quite annoying**.

Characters - Round 2

After round 1 we found out that the traditional compliance processes didn’t match with the agile mindset of the development team and their DevOps delivery cycle. Time to bring in some change, have the team really self organize around the security and quality requirements. And bring some new people to the team!

Sanjiv - the security champ of the dev team

We recruited Sanjiv to empower the team with all security related requirements. Sanjiv is a developer with a strong interest in information security. He will be the security champion of the team, keeping the team’s attention on building a secure product and also helping with covering the security requirements.
Thanks to Sanjiv, the team will no longer be interrupted and yelled at by Scary Steve.

Queenie - the developer with QA specialty

We brought Queenie to the team. Queenie is a developer with a special eye for quality. She is an all round developer, who will also bring the release and quality related topics under the attention (and also help address these requirements).

Since she is part of the team, Queenie will make sure that the team will no longer be delayed by the questions of Rachel.

Why this game?

In a traditional organization with a long history of calendar based software releases, quality is enforced in a very rigid way. Release managers and information security officers fire away their quality requirements. When dealing with long release cycles (waterfall-ish, project based) this could still work to some extent, but when you want to evolve to frequent deployments and release on demand, what will the impact be if these people keep firing their quality requirements from their trenches?

With this game we created a safe learning environment to experience exactly that, and also what the consequences are if you delegate (part of) these quality responsibilities to the development teams themselves.

Bug
Feature
Quality
Security
CI
Continuous
Interruption
CD
Continuous
Delay
QA
Quite
Annoying

What?

It's a card game. The name - CI-CD-QA - suggests this is about continuous integration, continuous deployment and quality assurance. The game indeed deals with these aspects, but in this case the abbreviations CI, CD and QA mean something different:

- Continuous Interruption
- Continuous delay
- Quite Annoying

This is meant to illustrate the impact of traditional CISO and release management on the desired Agile/DevOps flow of feature delivery. And that can be perceived as continuous interruptions and delays, which can be quite annoying…

What the image symbolizes

The image shows a long funnel, which could also be a pipeline that goes from wide to very narrow. The top of the funnel contains different bubbles of different types:

- features
- Bugs
- quality
- security

The funnel is so narrow that only drips come out every now and then.

Aim of the game

Experience how the interaction goes between all the involved parties and what the impact of this way of working is on the product delivery. This goes in 2 ways:

- everyone sticking to their role
- everyone working together to the common goal

Participants

- 1 PO - Paul
- 1 or 2 (better) dev team - Dave and Debby
- 1 CISO - Scary Steve
- 1 release manager - Rachel

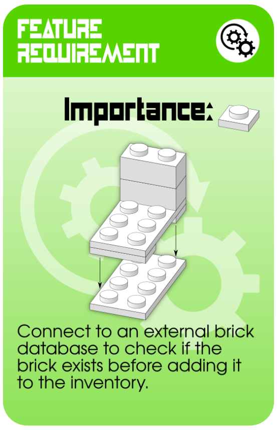
Types of cards

Each role has their type of cards they need to play:

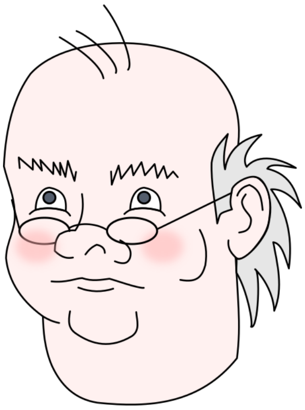
- The PO has feature requests and reported bugs
- The CISO has security requirements
- The release manager has quality requirements
- The developer has implementations of all the above

PO cards

Paul, the PO has 2 types of cards:


FEATURE REQUIREMENT
Importance:
Connect to an external brick database to check if the brick exists before adding it to the inventory.

BUG REPORT
Severity:



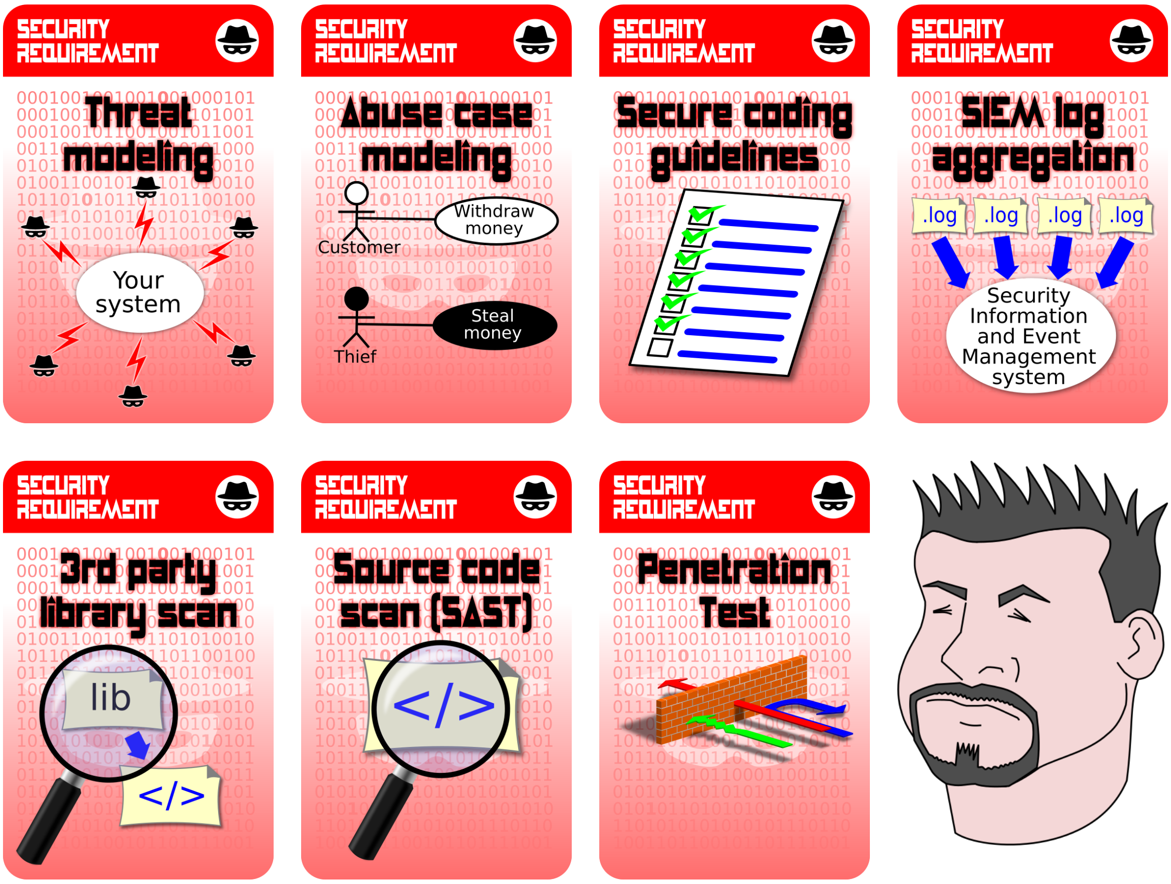
As you already know, BrixCo wants to bring a groundbreaking application to the market, not only to help their customers manage their bricks, but also to get better insight into the bricks and sets they own. Using this information the company can do direct marketing, make special offers about sets or bricks the customer might be interested in.

As product owner it is Paul's responsibility to put all the requirements and bugs he finds on the table. Feature requests come with an Importance value (1, 2 or 3 blocks high). The bugs come with a priority:

- yellow: low priority
- orange: medium priority
- red: high priority

CISO cards

Scary Steve needs to be sure that the information security of BrixCo is never compromised. Therefore he will make sure that his security requirements are met. This is what Scary Steve is constantly yelling about:

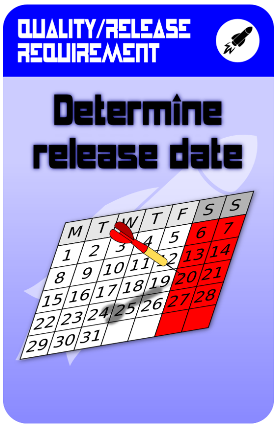


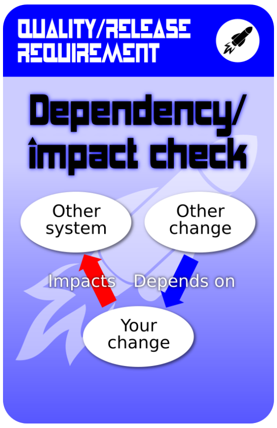
The CISO has to try to play all red requirements cards. In a first round the order doesn't really matter. At a later stage, with more experienced people, you can make things a bit more challenging:

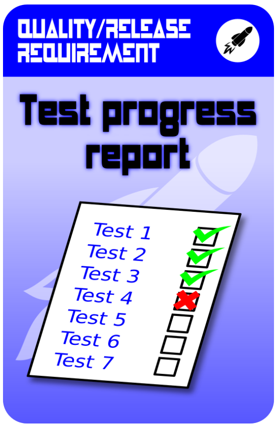
- the first security card to be played is the threat model card
- the last security card on table should be the Penetration test card
  This is typically the last security activity before a release

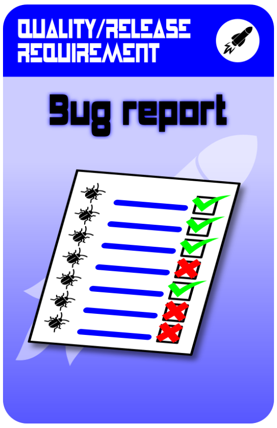
Release manager cards

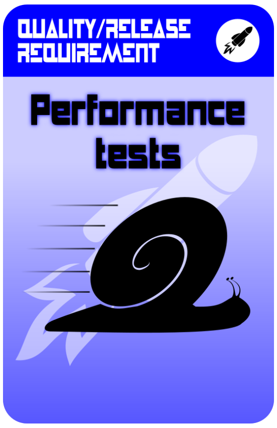
As release manager Rachel has to make sure that the developers don't compromise on quality when releasing a new (version of a) product to the market. because bad quality is bad for BrixCo's reputation. Therefore Rachel will put her requirements on the table:

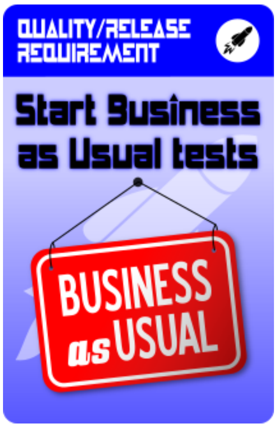


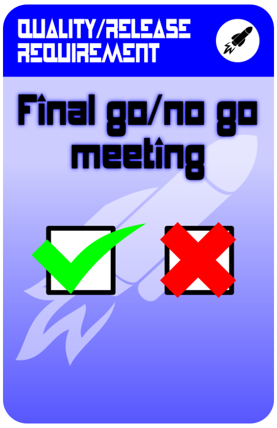


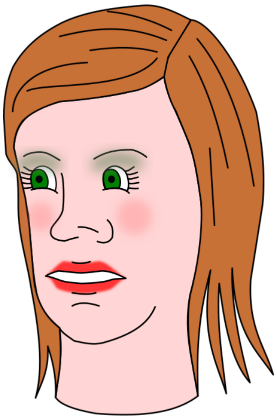










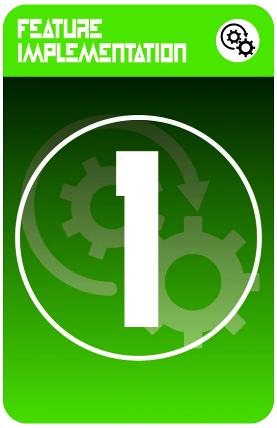


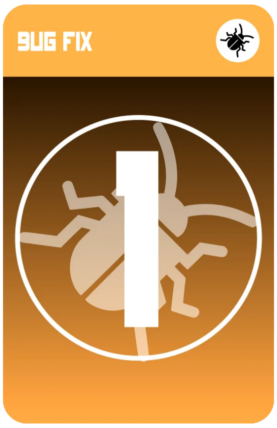
The release manager has to try to play all the blue cards. Initially the order of the cards is not important. At a later stage, with more experienced people, you can say that:

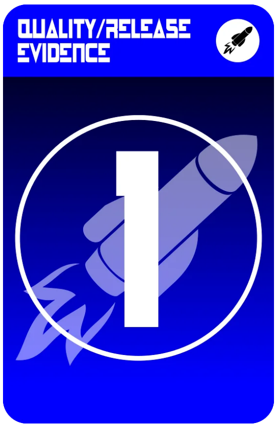
- the first card to be played is determining the release date
- the last card to be played is the final go/no go meeting

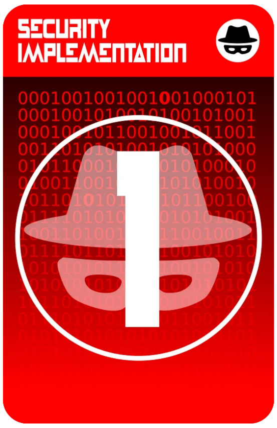
Developer cards

The developer needs to play cards that implement the above mentioned requirements, from all involved people:



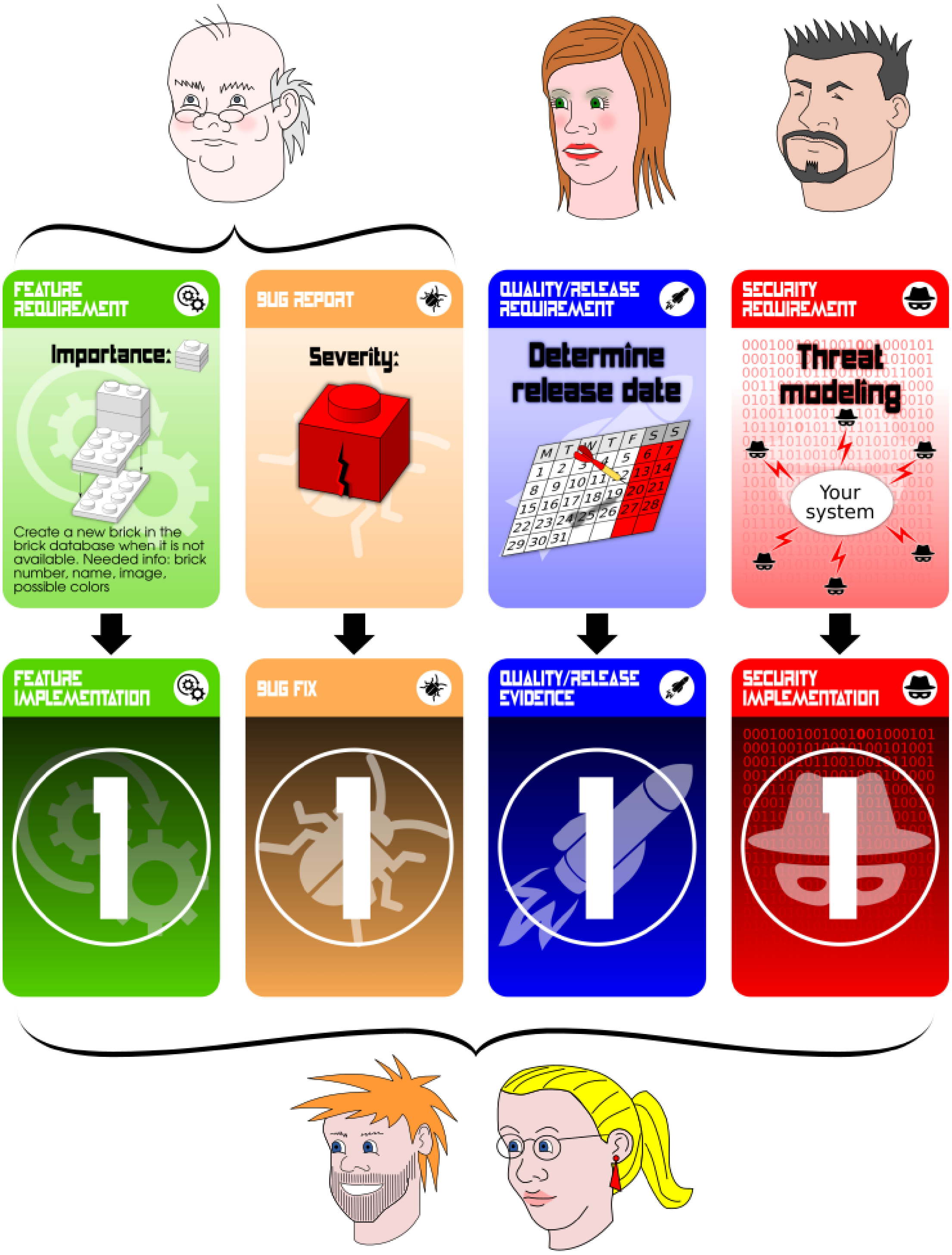




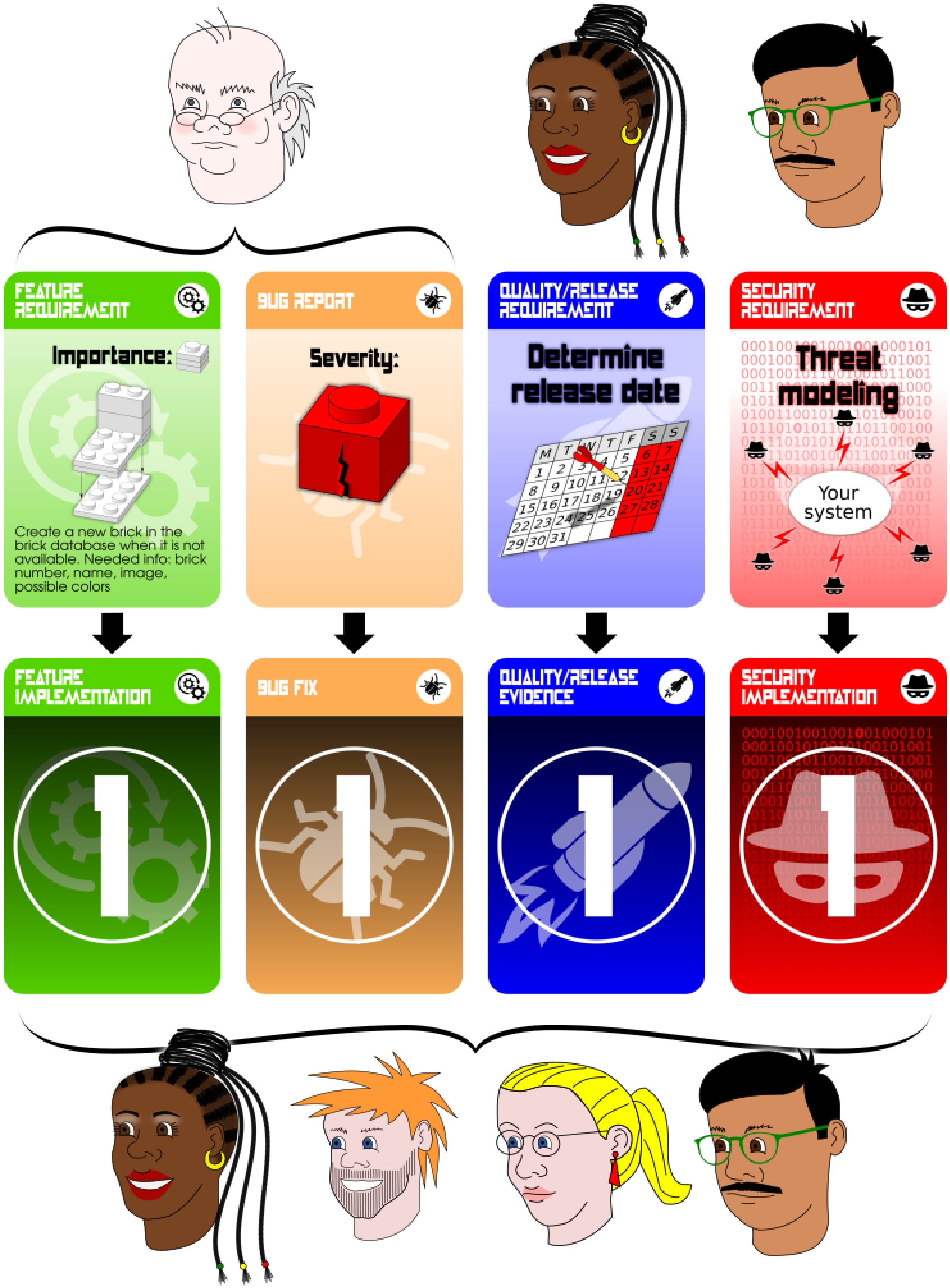


So each time someone puts a requirement on the table, the dev (team) has to try and cover it with a corresponding implementation card. That will determine the created value, the quality and the security of the application.

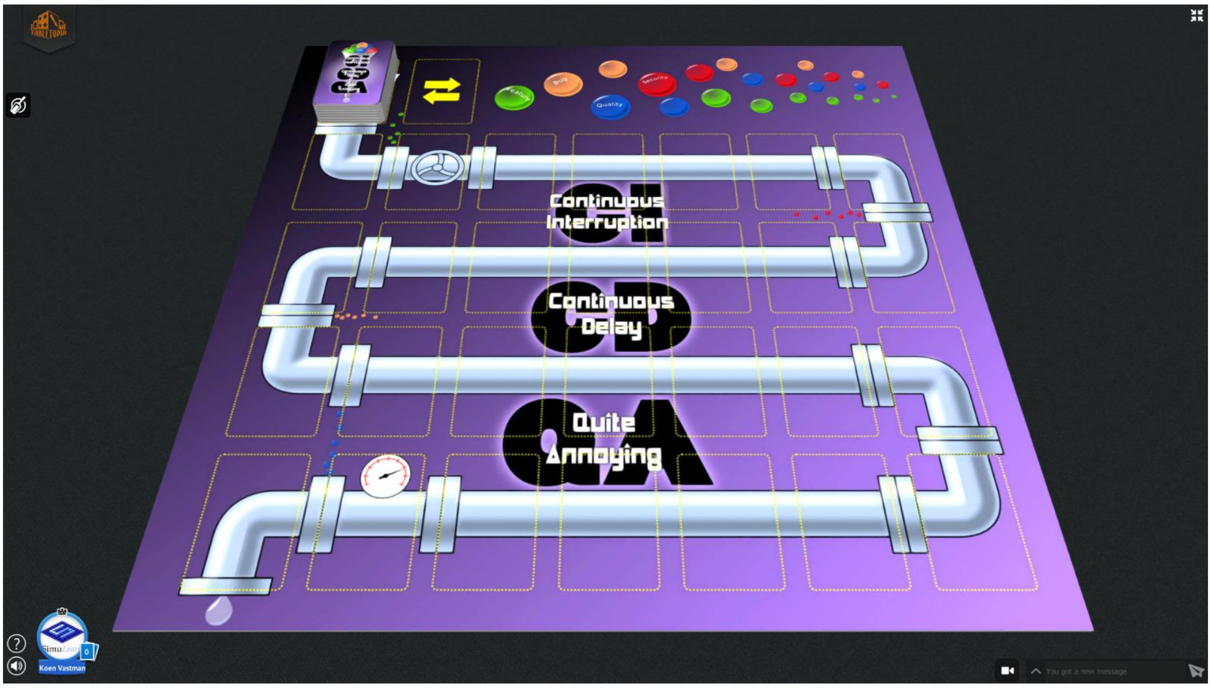
Who plays what - in round 1



Who plays what - in round 2



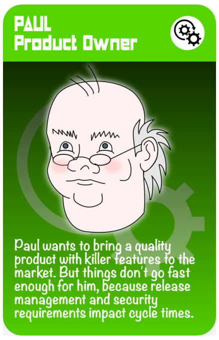
How the game goes



On Tabletopia we need to have a game board in order to test this card game, so I spent some time making a good looking game board that will also be part of the physical version.

Role assignment

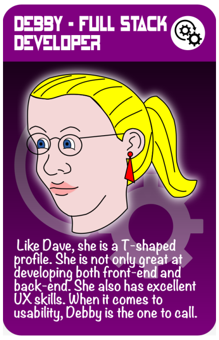
It all starts with the assignment of the roles. Each player blindly picks a card from the stack of role cards. The cards of Queenie and Sanjiv are left out in round 1. If there are only 4 players, you can leave out 1 of the developer cards - either Dave or Debby. The persona they picked from the deck will determine the role they play in the game:



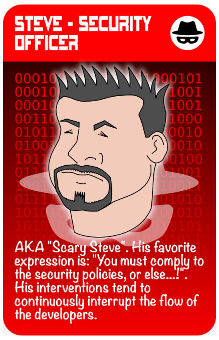
Paul wants to bring a quality product with killer features to the market. But things don't go fast enough for him, because release management and security requirements impact cycle times.

DAVE - Backend
Developer

Very skilled backend developer. His code is well written, covered by a lot of unit tests, and easy to maintain and extend. When it comes to database performance skills, Dave is the one to rely on.



Like Dave, she is a T-shaped profile. She is not only great at developing both front-end and back-end. She also has excellent UX skills. When it comes to usability, Debby is the one to call.



AKA "Scary Steve". His favorite expression is: "You must comply to the security policies, or else...!". His interventions tend to continuously interrupt the flow of the developers.

RACHEL - RELEASE
MANAGER

She does what she had to do: making sure that the developers deliver a quality release increment of the product. Her requirements often cause delays in the delivery process though...

In round 2 some cards will be swapped:

- The person playing Steve, the security officer, will take the role of Sanjiv instead, the security champion of the development team
- The person playing Rachel, the release manager, will take the role of Queenie, the QA person of the team

Playing the cards

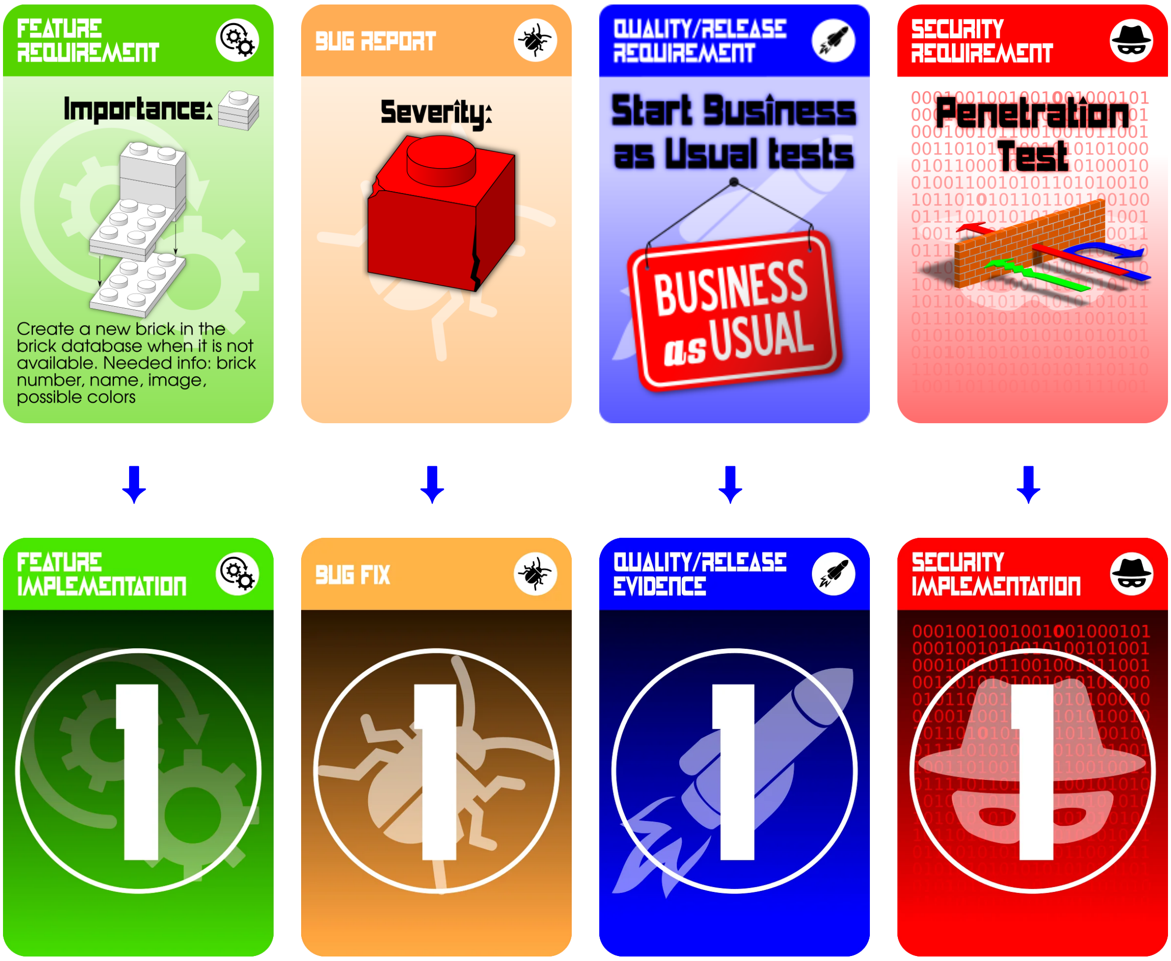
Each player gets 7 cards. How you deal these cards, is not important, as long as you make sure that the cards are well shuffled. The game starts with the PO who has to put a first requirement. No bugs, security or release requirements before a feature requirement got implemented. In the meantime, the release manager and security officer can swap a card they cannot use with a new one from the stack.
Play Go/No Go meeting or Penetration test at the earliest as the 8th requirements card on the table.

To keep an overview of what requirements have been played, it can be advisable to put all requirements of the same type on a row:

- one row for feature requests
- one row for bugs
- one row for security requirements
- one row for quality and release requirements

Implementing requirements

When the developers implement a requirement, they need to put a matching card on top of the requirement:



The numbers on these implementation cards don’t matter. They are just there to make a distinction between the cards.

If someone cannot play a card

If a player cannot put a card on the table, they can put all cards they cannot use back on the stack (below) and pick an equal number of cards from the top of the stack. If they picked a card they can play, they can do so, otherwise they need to wait until the next turn to swap cards again.

After the PO comes the release manager, putting their quality requirements on the table, 1 card per round. In a first round it does not matter in what order these cards are put on the table; in a later round it does:

- first card: picking the release date
- after card 8: final go/no go meeting
- inbetween: any type of quality requirement

Same here: if the release manager has no blue requirement cards in their hand they need to swap 1 card from their hand with 1 from the deck.

Third is the security officer, putting security requirements on the table, again 1 card per round. As with the release manager, the order of putting the cards on the table is not important in a first round. In a later round it does:

- first card: the threat model
- last card: the penetration test
- inbetween: any security requirement

And as for all players, if a developer cannot put an implementation card on the table, they have to swap 1 card from their hand with a card from the deck.

When does the game end?

The game ends with the first person that played all their cards. Then you can see:

- how much value has been delivered
- how many bugs are still open
- how secure the application is
- what the overall quality of the release is

You don’t reveal this from the beginning; you just say that the game ends when the first player has played all their cards and then you look at quality, value and so on.

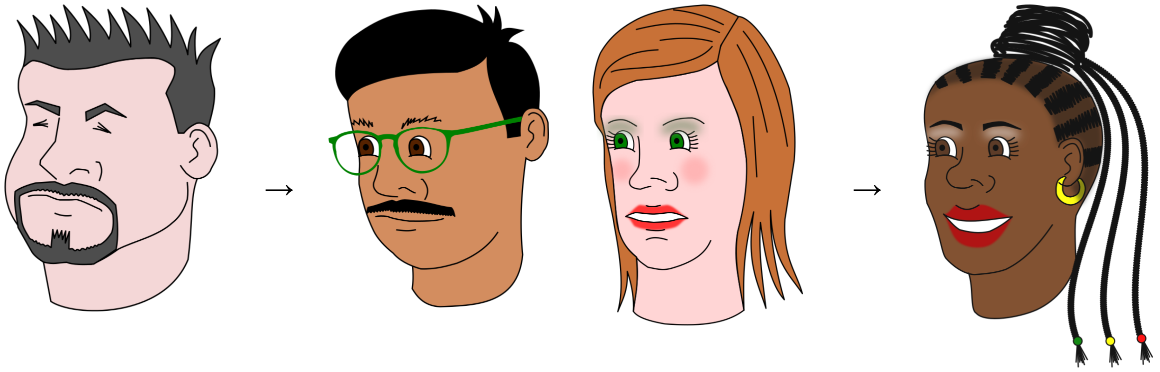
2 rounds to play the game

The first round - everyone sticks to their role

The PO tries to bring as many requirements to the table as possible. The CISO tries to bring in as many security measurements to the table as possible. The release manager tries to bring as many quality gates to the table as possible. In short: in this round their aim is to make sure the dev team cannot deliver. The winner is the first one who can put all their cards on the table.
It will feel as if they are bullying the dev team and cannot do anything else to contribute to a successful completion. They may still have a large stack of cards they cannot use.

The second round - collaborate

In round 2 we bring in 2 new characters:



Sanjiv replaces Steve for security. Queenie replaces Rachel for quality. Sanjiv and Queenie are part of the development team. This means that they can take up implementation activities and will also make sure that their respective requirements are put on the table. This means that no-one (except for the product owner) sticks to their role only, but helps each other out. It is like in a mature team where roles like release management and quality assurance are the responsibilities of the development teams themselves.

Alternative play

You could also choose to end the game once every player has played their cards. This will potentially maximize value and quality, because everyone gets the opportunity to play all their cards, potentially bringing more requirements and implementation cards on the table.

Thank you!

This game is a realization of SimuLearn in co-creation with Sven Cipido.

Special thanks goes to the people who tried out the game and gave their valuable feedback, that allowed us to improve the game:

- The Serious Gamers (Eddy Bruin & Jordan Gross)
- Jord Rengervé
- Gabrielle Bennefield
- JF Unson
- Jo Masraff
- Claudia Orozco-Gomez
- Angelina Dimova