SCRUMBLE

GAME RULES

AN AGILE GAME BY

PYXIS
agile know-how

NOTES AND ACKNOWLEDGMENTS

This guide was last updated on November 1, 2014. Please make sure you have the most recent version!

ABOUT THE GAME

Scrumble was created in 2014 in the whimsical spirit of Roman Trocherie, an Agile coach and Scrum Master at Pyxis Suisse. After all, if Scrum comes partly from a game, why not create a game on Scrum? That was the original idea.

Game design and experiments lasted for weeks, and Scrumble will continue to evolve based on your experiences. For any comments or suggestions, please contact Romain by email at rtrocherie@pyxis-tech.com.

LICENCE AND USER RIGHTS

The Scrumble game is completely free, playable by everyone, regardless of place and time.

However, we want to highlight that this required a substantial effort on the part of Pyxis Suisse's team and the final goal of the game, which is the promotion of an ideal operation of teams based on the principles of Scrum, should not be altered.

As such, the game and its different media are provided under licence (Creative Commons–Attribution – NonCommercial–No Derivatives 4.0 International). Therefore, you can share it by quoting authors, without modification and without financial compensation.

ACKNOWLEDGMENTS

You could not have read this or played Scrumble without the much appreciated participation of great and very playful people! They brainstormed relentlessly, threw dice thousands of times, dug until exhaustion... In short, a big thank you to the following individuals:

To the members of the Pyxis Suisse team (Christian Lapointe, Francesco Lomonaco, Gaël Luisier, Nedjma Saidani, Tremeur Balbous and Thomas Gibot) for their encouragement and for having served at the first guinea pigs!

To Pyxis Canada and **Pyxis Belgium** for their support during the design and development of the game.

And, of course, to you for playing Scrumble!

TABLE OF CONTENTS

INTRODUCTION

Scrumble is a game that is modeled on the roles, events, and artifacts of Scrum. Its aim is to present the operation and the precepts of Scrum in a team that wants to take over this development framework or reveal errors in the case of a faulty adoption.

The game allows, in a relaxed atmosphere, to highlight a number of problems faced by teams and, at the same time, to spontaneously find solutions.

The many difficulties are designed to test the players as actors creating real products. Disrupt, trigger reactions, develop individual and collective values required for the functioning of a team (and projects), this is the program for Scrumble.

We wish you great fun playing Scrumble and we hope it will allow you to bring improvements to your Scrum teams.

Let the game begin!

OBJECTIVES OF THE GAME AND DESIRED BEHAVIOUR

As noted in the introduction, Scrumble is a fun way to learn and consolidate Scrum when participants are reluctant to take the step themselves, to learn by using guides, or to appropriate it.

The great strength of Scrumble lies in its role of revealing dysfunctions in the team in a context that is conducive to transparency and interaction, and safe for everyone.

The rules described in this guide are ideally modeled on Scrum and may seem complex at first. However, they are not. You will realize it very quickly after the first sprint. Also, all rounds will be different, based on the players' creativity and the decisions they make. Together, they find new opportunities and sometimes problems where there are none! Of these challenges stem many lessons and Agile development:

- **Team spirit:** Everyone is actively involved in the game and is heard by others.
- **Creativity:** Without cheating, the game mechanics are exploited by all players.
- **Self-organization**: Initiatives are collective and spontaneous and adds rhythm to the game.
- **Transparency:** Everyone expresses oneself clearly and ideas are made understandable to all.
- **Respect:** Unnecessary tensions and disturbances are avoided and set aside of the game.
- **Conflict resolution:** Problems between players are resolved intelligently and smoothly.
- **Ability to prioritize:** The Product Owner targets the delivery of value, and the team contributes to it.
- **Commitment:** Everyone does their best to achieve a common objective.

PREREQUISITES AND PREPARATION

GAME COMPONENTS

Before beginning a Scrumble round, you need of course all the necessary materials and components. You will find them enclosed. They are ready to be printed and cut. Substitutes may be used instead of pawns and markers. For example, LEGO® bricks and characters are easy to integrate.

Here is what needs to be printed in color, cut, and glued:

- **The board, pawns, and markers:**
  Print these items on Bristol board (or stronger paper) in a large size (minimum A3). The A2 format is perfect for a large number of players. Then, cut parts keeping the small edge (shaded area on pawns and markers). Fold symmetrically the pawns representing the players and the debt, then glue their center so they can stand vertically (to facilitate movement on the board).

- **The cards:**
  Print on plain white paper or Bristol board in A4 format. By printing two-sided copies, the cards will have their cover in the back of their statement. If you want to save ink, you can print only the odd pages (you only have the statements on one side).

- **The set of User Stories:**

  If you want to play a game of Scrumble within a set context for the product to develop, print all pages except the last one.

  If you want to write your own User Stories, print only the last page. Then, together with the Product Owner, write all User Stories on the index cards obtained (one story per card).

  The User Stories can be printed on coloured paper of A4 format for an optimal finish.

The following items are also required: a dice (have several nearby in case of loss), blocks of Post-it® notes, pencils with an eraser at the end (preferably).

Accessories such as a whiteboard and markers can be useful, but they are not essential.

PREPARATION AND ORGANIZATION

You now have all the required items at hand. Also, you probably read through this guide either diagonally or entirely. Well done! Now you have to choose one or several teams to play with.

Regarding the organizational aspect, we recommend that you follow the following list for successful Scrumble rounds:

- Allow two hours for the game (that you will have previously planned with the team) and provide a pleasant setting. A room is essential to enjoy the game without being disturbed and without disturbing others.

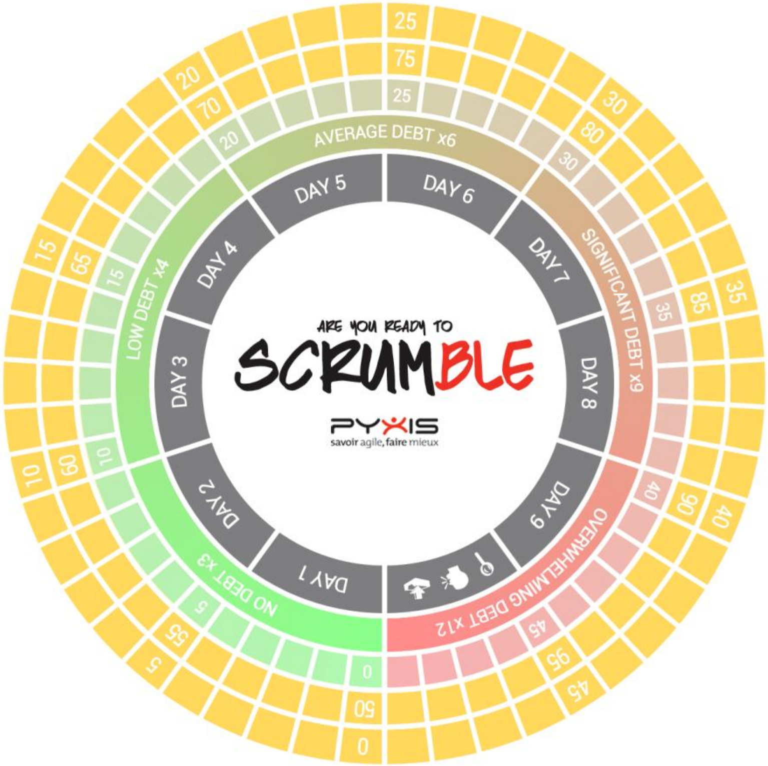
- Limit the number of players (between 5 and 11 maximum), including the Product Owner and Scrum Master. Communicate at least the introduction of the game previous to the activity.

- Make sure you have a Scrum Master in the group. This person must know Scrum sufficiently to be able to act as game leader.

- Have all the elements of the game in hand.

- If the game is played within the context of a real project, you can write your own User Stories, as indicated in the previous section. In this case, you can use an existing Product Backlog to get down to the pre-sprint phase. If the Product Backlog does not exist yet, we invite you to conduct a Big Wall workshop to develop it.

OVERVIEW OF THE GAME



On the left, there is the game board, starting from the outside towards the inside:

- Two circles for the tasks
- One circle for the points of debt
- One circle for the debt factor
- One time circle, in other words, of the different phases. The square following day 9 represents the review, the retrospective, and the planning for the next sprint.

The objectives marker and the players' pawn:

The different cards to draw:

PROBLEM
DAILY
REVIEW

The debt and time markers:

PLAYERS

As with the rules defined by Scrum, Scrumble uses three categories of players: Scrum Master, Product Owner, and member of the development team. To play, you need **one** Scrum Master, **one** Product Owner and **three to nine** people to act as members of the development team.

SCRUM MASTER

The Scrum Master is the game master. He or she does not act as player in the game. However, this role is a vital one as this person **conducts the activity and facilitates decision making at key moments** to ensure the smooth running of the game**.**

The game leader is sufficiently familiar with the rules detailed in this guide and also with those of Scrum. For an optimal experience, he or she must have the attention of all players and a minimum of respect from them. In addition, a game that is imposed only leads to boredom and a definite abandonment: the game leader must, therefore, not fall into the trap of organizing a simple game, but rather a gamified workshop.

If the Scrum Master (i. e., the team's real Scrum Master) does not master the game rules or is not able to organize the game, then this person will be invited to play as a member of the development team, and somebody else will be appointed game leader.

PRODUCT OWNER

No surprise, as the name suggests, the Product Owner is responsible for the product. This role is a special one: that of a director focusing on the delivery of value and that of a communicator knowing the field for which the product is intended.

During the game, the Product Owner acts as **decision maker** and as **intermediary between the development team and the users or eligible parties sponsoring the project**.

MEMBERS OF THE DEVELOPMENT TEAM

They are testers, analysts, junior or experienced developers, administrators, designers, trainees, etc. All are equal within the team (both the actual team and the one playing the game). They are self-organizing and they contribute to the development and incremental delivery of the product during all technical phases.

Therefore, the gameplay and the end of the game mainly rely on the members of the development team according to their qualities, skills, and interactions.

OBJECTIVES AND WIN CONDITIONS

OBJECTIVES

While the ultimate goal is a better appropriation of Scrum, it is advisable that the game leader refrains from giving the ins and outs of the game, that is to say the lessons to learn. Players will quickly discover them by themselves, and that's the whole point of the game!

In addition, expressing beforehand the expected lessons can easily distort the players' behaviour: desired interactions will not be experienced and the challenges will be less interesting. **Thus, the game leader will mention the real benefits only at the end of the game, illustrated by certain highlights of the game.**

Finally, to conduct the game, the players' success is quantified in ★ value points. These points are gained by performing 🟨 tasks sprint after sprint. The number of ★ value points must obviously be as large as possible at the end of the game. Success, symbolizing client satisfaction to receive product increments, is achieved according to the dynamics detailed further, which is filled with pitfalls as well as individual and collective challenges.

CONDITIONS FOR WINNING

There is no specific condition to win in this game. That's right!

Indeed, the players' success is evaluated in the completion of the product in a reduced number of sprints. The real success, as you may have understood, lies in all the learning achieved during the game and in its future application to a real context!

The only possibility of failure is the capitulation of all players without any lessons learned. I reassure you immediately; this is a highly unlikely situation.

DURATION

The game leader chooses a number of sprints to achieve according to the time allocated and the number of players.

It takes approximately 15 minutes to clearly explain the game rules and another 15 minutes for the Product Owner to present the product to develop and its context. Allow 5 minutes for the planning, 2 minutes per player per sprint and an additional 10 minutes per sprint for the review and the retrospective. It is possible to play in pairs (similar to pair programming!) in order to save time without compromising the quality of the game experience.

PHASES OF THE GAME

PRE-SPRINT

Once the game leader explained Scrumble and gave the game rules (refer to the attachments for the breadcrumb), which took just a few minutes, the Product Owner presents to the team the product they will have to create during the game. The Product Owner is the only person who gives the vision and the Product Backlog (that he or she prepared in advance) as well as the user stories taken (or not) from the sets of User Stories provided.

The Product Backlog is contains all the User Stories populate the sprints.

Whether it is a real product or a dummy, it is important that the players like the idea and theme of the product to ensure their participation.

The Product Owner provides the team with 15 User Stories (use cases, functions) written on sticky notes. Here is what they should look like:

User Story # Depends on another User Story

As [WHO?],
I want [WHAT?].

Value: Size:
[1 to 5 ★ points of effort] [Size?]

The Product Owner allocates 45 ⭐ value points **to User Stories**, i.e. from 1 to 5 each (an average of 3 ⭐ value points per User Story). A User Story with more ⭐ value points is more interesting and satisfying to users when they deliver it.

You can note the **Size** item on the sticky note.

Once the User Stories have been presented and understood, **team members have to estimate the relative complexity (size) of each User Story to be implemented.** When this is done, the team has a number of 🟨 tasks to carry out (or objectives to achieve) during the sprint.

Non-exhaustively and using relativity, developers then provide their estimations by sorting the User Stories according to the following table.

Priority and complexity	1–XS	2–S	3–M	4–L	5–XL
High					
Average					
Low					

Do not hesitate to post them on a dedicated space of the play area so they remain visible during the entire game.

The relative complexity or size (allocated the t-shirt way) allows to keep in mind that an exact estimate of effort and work load is never possible. All human beings are, by nature, bad in absolute estimating. However, since there are many User Stories to be executed per sprint, the error is generally reduced. Moreover, it will give food for thought to those whose estimates are often done in person-days and who take these estimates for granted.

Here's for the little story! It is now time for the first sprint and its planning. As needed, use the 💰 **technical debt** marker and place it to obtain 15 points of debt and a factor 4.

SPRINT

In chronological order, a sprint comprises a planning session, 9 development days (9 turns), a review session and a retrospective.

Note: The day without development out of the 10 working days is devoted to the planning and review sessions as well as the retrospective. Somebody will most probably ask the question, so you will know what to answer!

SPRINT PLANNING

During the sprint planning phase, team members choose the number of 🟨 tasks they will carry out over the sprint.

First, **the Product Owner chooses the product's priority and high-value User Stories** (user-oriented focus) that have possible dependencies. Developers can give suggestions to the Product Owner, but he or she is the one who has the last word.

The chosen User Stories determine the sprint goal. For the first sprint, it is recommended to take a maximum of 3 User Stories to begin smoothly.

The sprint goal is broken down into mini-objectives that, once achieved, reward the team with the number of ⭐ value points indicated on the User Stories when related 🗒 tasks are completed.

The number of 🗒 tasks is obtained with the following formula based on the 💰 technical debt (its impact is explained in the section called **Your nemesis: the technical debt**) and the 👤 number of players in the development team (excluding the Product Owner and Scrum Master).

Debt and complexity	1–XS	2–S	3–M	4–L	5–XL
💰 x 3	3 x 👤	6 x 👤	9 x 👤	12 x 👤	15 x 👤
💰 x 4	4 x 👤	8 x 👤	12 x 👤	16 x 👤	20 x 👤
💰 x 6	6 x 👤	12 x 👤	18 x 👤	24 x 👤	30 x 👤
💰 x 9	9 x 👤	18 x 👤	27 x 👤	36 x 👤	45 x 👤
💰 x 12	12 x 👤	24 x 👤	36 x 👤	48 x 👤	60 x 👤

On the corresponding User Story's sticky note, we have to write the number of 🗒 tasks to carry out. Note that this quota can be later affected by bonus-malus or if the objective is not achieved during the sprint and the team has to continue the work during the following one.

Then, for each objective, we have to place a coloured marker on the corresponding number to be achieved located on the circle of 🗒 tasks. We also have to place a gaming pawn of the same color on 0. **The User Story will be done and deliverable once both the pawn and marker are on the same square.** If the objective of the User Story exceeds 100 🗒 tasks, distribute several markers with the same color on the circle; you will have to collect them all later on.

Let the development begin and start day 1!

DEVELOPMENT DAYS

It is time to carry out the different 🗒 tasks. **Each member of the development team (excluding the Product Owner and Scrum Master) rolls the dice in turn after choosing to execute 🗒 tasks or to reduce the 💰 technical debt.**

If the player chooses to reduce the 💰 technical debt, he or she moves the appropriate marker backwards on the circle of debt (inner circle). For the execution of 🗒 tasks, the player moves his or her pawn forward on the outer circle (light yellow) according to the number obtained when rolling the dice.

The player chooses the option that he or she believes will give the most value to the sprint or product. It's your assessment!

In both cases:

- **If the player rolls a 1**, he or she can roll the dice again (no more than twice).
- **If the player rolls a 6**, he or she picks a 💣 problem card, reads it, and presents the story to the team. The Product Owner is also invited to give his or her comments.

The 💣 problem card is most of the time a complication to the game. The team has to act together to overcome the difficulty, not only the player who picked the card.

Each new day, except on day 1 or at the review, one of the players picks à 💬 daily card. It can be a challenge for the team to face, a question with an incentive, or an anecdote about the Agility.

When a turn is over, we go to the next day, up to the review.

If all 🟨 tasks to carry out during a sprint are done before the end of the day, the team chooses one of the following options:

- **Add one or several User Stories to the current Sprint Backlog**; thus delivering more value during the sprint. In this case, we add a pawn and markers and we interrupt the current turn (day) because of the effort deployed and we go to the next day.
- **Pay** 💰 **technical debt** for the remaining of the sprint. Why not?
- **End the sprint.** This option is only possible if there is no User Story left in the Product Backlog. We conduct a post-mortem of the game by going directly to the last review.

REVIEW

When reaching the Review square, the Product Owner inspects what has been done during the sprint (events and score). He or she assesses the progress of the product. He can also congratulate, encourage, and give constructive advices to the team. This phase must not exceed 5 minutes.

The Product Owner picks one 📍 review card per User Story completely done during the sprint. Someone reads the cards out loud one after the other. The instructions on the cards can impact the game now or later on.

Regarding User Stories that are not done because of technical or functional activities that have to be reviewed and redone, **the team gets a malus**. This malus gives as many 💰 technical debt points as there are groupings of 5 🟨 tasks not completed during the sprint. The User Stories with markers already on the board are not impacted (i.e. their number of 🟨 tasks do not change). The handicap will only be applied to the new User Stories being developed and to the effort required later in order to reduce this 💰 technical debt.

Finally, for the last part of the review, the game leader calculates the number of points achieved since the beginning of the game. He or she is the one who gives to the team the ⭐ value points per User Story done. Bravo!

It is now time for the retrospective!

RETROSPECTIVE

This specific moment is dedicated to the participants. The game leader gives between 1 and 2 minutes to each player to express what he or she liked, disliked, and hated during the sprint and also what he or she intends to implement during the next sprint. Initiatives are highly encouraged.

During this phase, the dynamics between the players' practices and values must be more and more Agile as the sprints go by. It is supposed to be the most revealing and formative phase. Of course, the last retrospective of the game can be longer.

YOUR NEMESIS: THE TECHNICAL DEBT

It is your enemy. It takes several forms... It is everywhere...

The nightmare of many Scrum teams is the 💰 technical debt. Whether explicit or not, it represents a due towards users and eligible parties when User Stories or 🟨 tasks do not respect the definition of "Done".

Thus, the 💰 technical debt will disrupt the game at different moments (for instance, when there is uncompleted work at the end of a sprint). It causes more difficulties when it keeps increasing. A new stage is reached every 10 points.

At the beginning of the game, the default handicap is 0 with the marker on 15 points. Then, it can evolve positively or negatively, based on the difficulties encountered and decisions taken by the players.

Here is a summary of the 💰 technical debt:

Points of debt	0 to 9 💰	10 to 19 💰	20 to 29 💰	30 to 39 💰	40 to 49 💰
Debt factor	x 3	x 4	x 6	x 9	x 12
Handicap	- 25%	0%	+ 50%	+ 125%	+ 200%

The 💰 technical debt's upper limit is 49. At this point, the situation will be unbearable. Let's hope you will be able to avoid it!

END OF THE GAME

The game ends after the last retrospective or as soon as the team gets stuck or has had enough (it is very rare, but if it happens, collective thought needs to be given about the obstacles encountered during the game—this discussion being facilitated by the Scrum Master).

The game leader gives the highlights of the game and finally reveals the game's real objectives. The total number of ⭐ value points can be calculated for fun. All participants should leave with a smile on their face and should be full of ideas. The final score is not as important as what has been learned during these few hours before beginning a new project or going back to the current project.

Did you perform outstanding work? Has the 💰 technical debt been mastered and defeated? What are the strengths and weaknesses of your team? Is this game a real one? So many questions to be answered and addressed to your colleagues...

Congratulations, you won!

APPENDIXES

ROLE-PLAYING: PRE-SPRINT

If you do not have the soul of a rolist or if you are lacking inspiration, here is an introduction that can be used for the pre-sprint. Do not worry, the rest will come easily!

> Hi everyone,
>
> As you probably know, **[Business Client]** contacted us a few days ago and asked us to develop a super product that will **[do something very nice]** and that is absolutely new in this field. The clients and eligible parties of **[Business Client]** became closer to **[name of the Product Owner]**, the Product Owner, who told them about your Agile skills and competencies which allow you to deliver great functions frequently and progressively.
>
> I am **[name of the Game Leader]**, the game leader. As Scrum Master, I will assist you during this adventure. As you, developers, know, delivering fully operational functions within two weeks isn't easy! And having fun doing it, yes, it's possible!
>
> During the next few hours, we will have time to explore this further. I hope that you will play the game and have fun doing it. Now, let's hear what **[name of the Product Owner]** has to say about the product to be developed. We will begin to address the challenge afterwards.
>
> Enjoy the game!

FREQUENTLY ASKED QUESTIONS

Here is useful information that has not been addressed so far.

- **One or several players have to leave during the game. Is it the end of it?**

  Even though it is undesirable, any player can leave the game at any time. It can also happen during a sprint of a real project; thus, possibly jeopardizing the Scrum team.

  If this situation occurs, the current sprint continues according to the rules, and the outgoing player can donate his or her permanent bonus-malus. Before beginning the next sprint, carefully reconsider the number of players.

- **Scrumble lacks competitiveness. How can we rectify the situation?**

  Even though the main objective of Scrumble is to develop team spirit and values, it is possible to add a little bit of competitiveness. If there are many competitors in your team or if you have a large number of players, you can consider playing many games simultaneously with teams of equal size. Then, the winning team will be the one who finishes its Product Backlog first (you have to use the same set of User Stories for all teams). The Product Owner and game leader can supervise the teams in different phases well remaining organized.

- **Do two teams of different sizes have an equal chance to progress and succeed during the game?**

  Absolutely! The game difficulty, notwithstanding unforeseen events, has nothing to do with the size of a Scrum team.

The difficulty does not depend on the number of players, but on the decisions they make. Two teams of 5 and 10 players respectively can finish a Product Backlog in as many sprints and days if they make similar decisions.

VARIOUS ANECDOTES

This game has not been developed with Scrum. It is Personal Kanban that has been used.

Even though certain concepts are similar to those of serious games, the idea of Scrumble came to the creator's mind when he took the Gamification course. This course is given by Kevin Werbach on a regular basis and is free of charge. We recommend it to anyone who is curious and likes to play games.

Would you like to know the average progress of a developer per day during the game? If no bonus-malus is applied, it is 4.08. This could be useful information for your predictions.

What is the average debt per player and per sprint? Well, it is too complex to calculate! Just remember that you will accumulate fewer debts by demonstrating good judgment.

SCRUMBLE IS PROVIDED BY

PYXIS
agile know-how

/campus

Learn about **Agility** and **software engineering practices**

/consulting

What Agility brings to **diagnostics, strategy, project management** and **coaching**

/studio

Ask for **custom software solutions** to our **expert developers**