

## Background

Welcome to MagicSoftware Inc. We are delighted that you are joining us as a new employee today. As you know, we build software for the digitalization of reporting. A year ago, we introduced „agile“ as our working mode. You will certainly get used to it quickly. Perhaps now it is best to meet your team:

Harald 🧑 is your team lead. He used to be a developer here before he took over the management role. Accordingly, he gets give you a helping hand when needed.

Mathilde 🧑 is part of the quality assurance and your contact regarding everything to do with testing. Sabine 🧑 is the customer manager. That means you don't have to deal with customers yourself. She has your back. At the same time, however, we have a shortage of staff, so Sabine has to help out elsewhere. Please be patient with her.

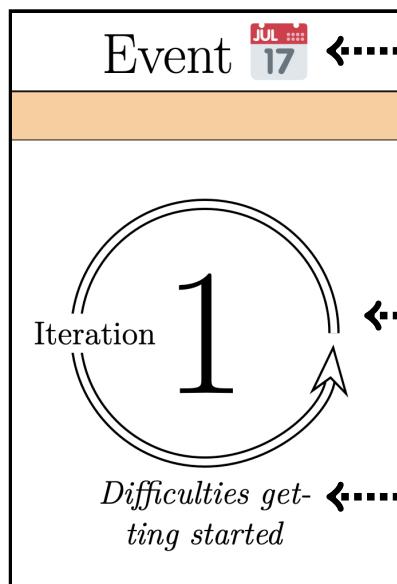
Mark 🧑 is the customer. You won't be in direct contact with him, but it's good to know who you're actually working for.

The project will be completed in nine iterations. Some things have been completed by your predecessors. Others have been left undone. But we are sure you will manage. Good luck!

## Game Elements

- 9 event cards
- 39 ticket cards, including:
  - 27 features
  - 12 bugs
- 12 double-sided tokens for technical debt
- 20 bears as tokens for story points
- 1 game board

## Event Cards



**Card type:** Events are unforeseen incidents that occur during the project. You will find out whether they have a positive or negative effect during execution.

**Upcoming Iteration.** There are nine iterations in total.

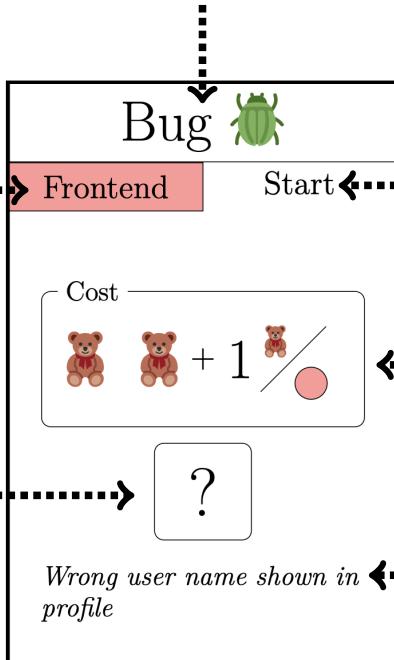
**Title:** Hint of what to expect from the iteration.

## Tickets

**Card Type:** Shows whether it is a feature or a bug. The difference between the two is that only features earn you points, but they also generate technical debt. Bugs do not.

**System part:** Indicates whether the ticket belongs to the frontend or backend.

**Effect hint:** When processing a ticket, you execute the effect on the back. The effect hint shows whether you have to expect a positive ( $\uparrow$ ), negative ( $\downarrow$ ) or random (?) effect.



**Start marker (optional):** Shows whether the ticket is on the game board at the beginning of the game.

**Cost:** Amount of story points you have to use for the ticket. The fixed cost are the amount of bears on the left side. Additionally, you have to use one storypoint for each technical debt in the corresponding system part.

**Title:** A description of the ticket.

## Technical Debt

Technical debt makes it harder to add more functionality. How much technical debt you have is shown by the corresponding tokens. Frontend debt looks like this: . Backend debt like this: .



## Story points

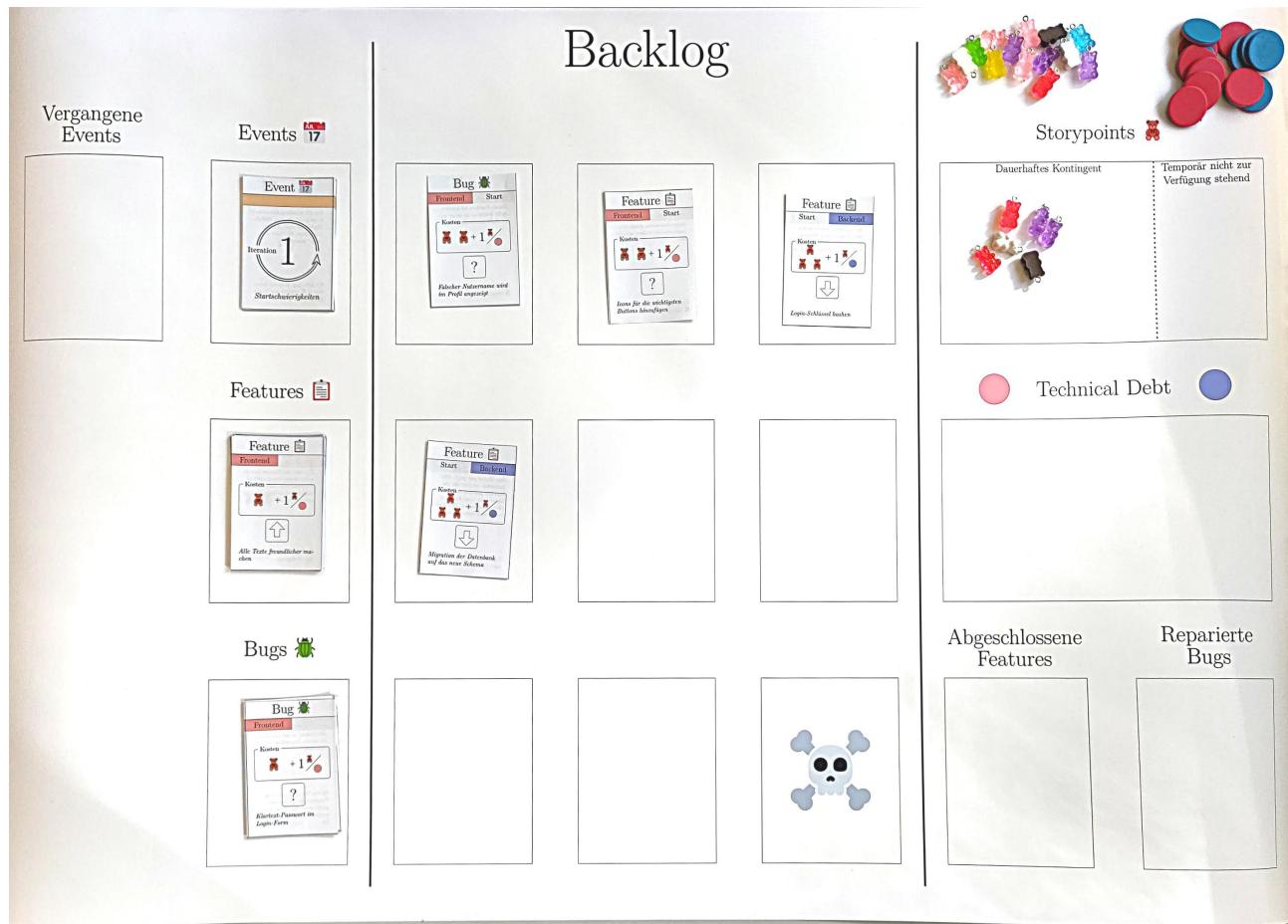
Storypoints are a unit of measurement for the effort you have to put into processing a ticket. They are displayed in the text with the following symbol: . You have a certain quota of storypoints available per sprint. They regenerate at the end of the sprint. The amount of storypoints available can change due to effects. The physical tokens are these bears:



## Setup

We have already set the game up. The instructions for that are below. Please check that everything corresponds to the instructions. If anything is different, please let us know.

- Sort the events according to iteration. The top card is for iteration 1, the bottom card is for iteration 9. Place them on the event stack.
- Pick out the three features with start markers and place them in the backlog.
- Shuffle the remaining features and place them on the feature stack.
- Pick out the one bug with a start marker and place it in the backlog.
- Shuffle the remaining bugs and place them on the bug stack
- Place the 13 technical debt tokens next to the board.
- Place six story points in the story point field. This is your permanent quota until further notice.
- Place the remaining 14 story points next to the playing field.



## Gameplay

The game consists of successive rounds or iterations. An iteration consists of the following steps:

1. **Event:** Draw the top card from the event stack and execute its effect.
2. **New feature:** Draw the top card from the feature stack and place it into the backlog with the cost calculation facing up.
3. **Planning the iteration:** Choose one or multiple actions for the iteration. The possible actions are:
  - **Choosing tickets:** Select one or multiple tickets from the backlog for processing. Put the needed amount of story points on the corresponding ticket. The amount of needed story points is the sum of the fixed price on the left side of the equation and the amount of technical debt in the ticket's system part. Caution: If you want to refactor the frontend or backend in this iteration, you cannot select tickets from that system part.
  - **Reduction of technical debt / refactoring:** For each technical debt you want to remove by refactoring, you need to use one storypoint. Put the story point on the technical debt to remove. Caution: If you refactor in the frontend, you cannot process frontend tickets. The same applies to the backend. You can refactor the frontend and backend at the same time but then you cannot process any tickets.
  - **Drawing additional feature(s):** Draw one or more features from the feature stack and put them in the backlog with the cost calculation facing up. This costs one story point for each additional feature.
4. **Executing the iteration:** Process everything you planned for the iteration:
  - Remove technical debt marked for refactoring
  - Process the selected tickets by taking one ticket at a time, turning it around and executing its effect. The order in which you process the tickets is up to you.
5. **Increasing technical debt:** If you processed at least one feature in the frontend, add exactly one frontend technical debt. Multiple frontend features do not result in multiple frontend debt. The same applies to the backend. Processing bugs does not lead to an increase of technical debt.
6. **Checking the end condition:** Check whether one of the end conditions is met. If so, the game ends. Otherwise, prepare the next iteration.
7. **Preparing the next iteration:** Put the storypoints used in the planning phase back into the story point field. Caution: The amount of available story points can be altered by effects. Sometimes you will have more or less story points available for the next sprint. The effects make clear what to do in that regard.

## End of game and points

The game ends in one of two ways:

1. At the end of an iteration there are nine or more tickets in the backlog. You lose.
2. You complete all nine iteration without losing. You win.

If you lost the game, you have 0 points. If you won, each processed feature gives one points. Fixed bugs do not grant points.