

Let's try...



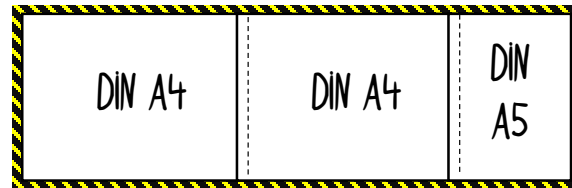


- The goal is to deliver approx. 100 Cards on the table in ascending order in the working area.
Doublettes are not allowed.
- We will play several Sprints (a 60sec), then debrief

Setup



- Playing with the printed set: Print one set of 130 cards for each group.
- Playing with „The Mind“ cards, you need 2 set for one group:
 - Use all cards with value ending on 4 and ending on 7 from one deck and add them to the second. Add also randomly 10 additional cards from the first to the second. Use the 130 Cards deck for the group.
- Shuffle each deck
- Each group will play in a different refactoring modes. Place Mode instructions and a different protocol-sheet on each table.
- Setup Working area



- Group of at least 4 active player, better 7 active player, rest observer

Roles



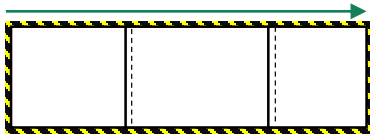
- 4 **Developer**: Take cards on his hands. Play cards into working area.
- 1 **Quality Assurance (QA)**:
 - Has to notify bugs
 - Can talk anytime and point everywhere
 - Can lift cards to check the number but must not move **any** card
- 1 **Product Owner**:
 - Arrange Features (5 Cards on a stack per feature, for the developer-team)
 - Count & Protocols # of features pulled by the dev-team each sprint
 - You can treat all these features as committed in the planning, the work from the PO here is just helping to keep the game table clear of too much cards
- 1 **Timekeeper**:
 - Checks time for Sprint 60sec,
 - Gives a hint 15 sec before Sprint ends
 - Takes time after sprint, how long it takes to check for bugs
- n **Observer**:
 - Have a look a speed over time, decisions, discussions, cheating, strategies etc.



Developer: How to play



- Each feature card you get, you have to keep in your hand and play from hand into the gaming area card by card
- All cards in the gaming area has to be in an ascending order (from left to right)



See page „Bugs“ for further details

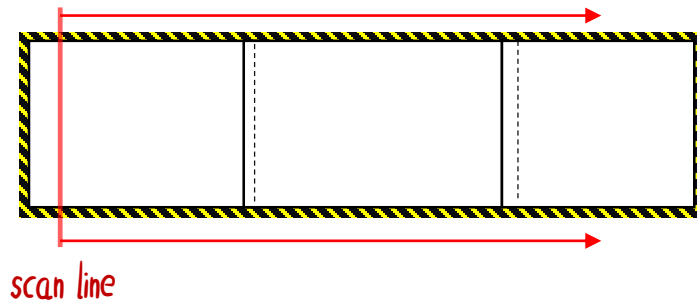
- Normally for the Developers it is **not allowed** to touch or move cards placed on the table beforehand. Only your own last played card is movable. (Perhaps if QA gives a hint)
- In some rounds/groups there are refactoring budgets available, which changes that rule above: Playing a refactoring card enables one developer to move/remove/reorder/cleaning up to 10 (I trust you not to cheat) cards on the table. If you need more you have to play another refactoring card. The played card is valid for one specific developer.



Gameplay each sprint



- If Sprint (60 sec) is over NO card must be played/moved
- The PO declares which refactoring mode this sprint is valid and gives budget (if available) to the team
- After each sprint:
 - Count the number of cards not played (on the developers hand) & protocol them
 - Timekeeper stop the time for the next activity. QA figure out how many bugs are in the working area using the scan line method. If necessary you can lift cards to check the card number but never move any card!
 - Protocol the # of bugs and the time it consumed to check.
What is a bug: See bugs-rule.





Change request

- From the second sprint the PO places some change request cards between the stack of features
- Each card is valid for this specific sprint. This means all cards on the board that has the requested value to change, has to be removed from the board. Place them outside the working area on the table
- For a change request for the developers it is allowed to touch (but not move) the other older cards and remove the requested cards (without using refactoring)
- Every new card with the mentioned value coming this sprint has also be removed/placed outside the area.
- At the end of this sprint every card in the gaming area with a mentioned value is counted as one bug.
- After the sprint you remove the change request card from the table, it is no longer in effect, the removed cards stay out of the area, but new cards could now be placed again as before.

Bugs



- Cards out of the working area or touching the frontiers
- Doublettes: Cards with the same number, Developers has to play those cards completely outside the area
- Cards not in the right ascending order (Scan the working area with a vertical „scan line“ from left to right. Every time you found a card with a lower card value then the last scanned card, it is a bug. Every card will be scanned only once.



34 ok because reached by the scan line first
11 Bug because lower than 56
11 a second bug because out of area
76 ok, but...
2nd 76 is a bug because it is a doublette

- Every not fulfilled change request for this sprint (Change request ar only valid for that sprint, not for a later one)

Refactoring mode: Feature, Feature, Feature! No Refactoring



In this group/round:

- No refactoring is allowed, we have to produce features, no time to clean up.
- Bugs stay in this constellation and will be counted every sprint again



Refactoring mode: Refactoring budget

In this group/round:

- Playing a refactoring card enables a developer to move/remove/reorder/cleaning up to 10 (I trust you not to cheat) cards in the working area. If you need more than 10 cards you have to play another refactor card
- Refactoring cards can of course be played to remove bugs identified at the end of the sprint before.
- The PO offers the team a budget for refactoring (changing older cards) for each sprint at the beginning of the sprint
- Not used refactoring cards comes back to the PO's budget at the end of the sprint

Refactoring mode: Unlimited refactoring



In this group/round:

- For the Developers it is allowed and suggested to touch cards placed on the table beforehand anytime.
- Refactoring can of course be used to remove bugs identified at the end of the sprint



Sprint	Refactoring Mode	# Features pulled	CR's	# Cards on hand WIP	# Bugs	Time for bug search
1	No refactoring					
2	No refactoring		CR-1@3rd			
3	No refactoring		CR-2@2nd CR-3@4th			
<hr/>						
4	Unlimited refactoring		CR-4@2nd CR-5@4th			
5	Unlimited refactoring		CR-6@3th			
6	Unlimited refactoring		CR-7@2nd CR-8@4th			

Debrief

Debrief



Sprint	Refactoring Mode	# Features pulled	CR's	# Cards on hand WIP	# Bugs	Time for bug search
1	No refactoring					
2	Unlimited refactoring		CR-1@3rd			
3	Unlimited refactoring		CR-2@2nd CR-3@4th			
Debrief 4	Unlimited refactoring		CR-4@2nd CR-5@4th			
5	Unlimited refactoring		CR-6@3th			
6	Unlimited refactoring		CR-7@2nd CR-8@4th			
Debrief						



Sprint	Refactoring Mode	# Features pulled	CR's	# Cards on hand WIP	# Bugs	Time for bug search
1	No refactoring					
2	No refactoring		CR-1@3rd			
3	Refactoring budget 1 Cards		CR-2@2nd CR-3@4th			
4	Refactoring budget 2 Cards		CR-4@2nd CR-5@4th			
5	Refactoring budget 4 Cards		CR-6@3th			
6	Refactoring budget 4 Cards		CR-7@2nd CR-8@4th			

Debrief

Debrief

Sprint		# Features pulled	CR's	# Cards on hand WIP	# Bugs	Time for bug search
1						
2						
3						
4						
5						
6						

