

# Code Monkey "Mike"

1

2



Employee - DEV

2

## When played:

you may return one of your employees to your hand.  
This ability can't target outsourced developers.

---

*"Good code documents itself?" So that's why we have to write so much documentation...*

# Code Monkey "Mike"

1

2



Employee - DEV

2

## When played:

you may return one of your employees to your hand.  
This ability can't target outsourced developers.

---

*"Good code documents itself?" So that's why we have to write so much documentation...*

# Ambitious Apprentice

1



Employee - DEV

0

Playing and transferring knowledge cards on this developer costs (2) less.

---

*Testing in production works better than coffee!*

# Ambitious Apprentice

1



Employee - DEV

0

Playing and transferring knowledge cards on this developer costs (2) less.

---

*Testing in production works better than coffee!*

## The Assertive CTO

1 2 3 4 5



Employee - DEV

4

### When played:

draw a card. If the drawn card is a developer:  
you may play it this turn for (0).

---

*Premature optimization is the root of all evil.*

# Copy Paste Developer

1



Employee - DEV

2

## When played:

your other developers get (1) burnout point.

---

*Ctrl C + Ctrl V. Commit. Done!*

# Copy Paste Developer

1



Employee - DEV

2

## When played:

your other developers get (1) burnout point.

---

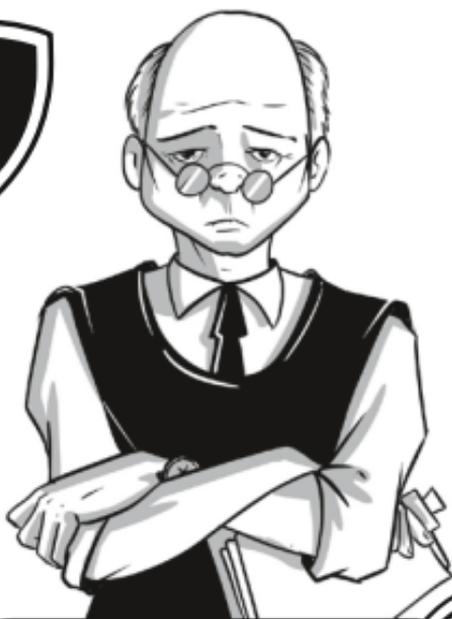
*Ctrl C + Ctrl V. Commit. Done!*

## Downer Dave

1

2

3



Employee - DEV

1

### When played:

add (1) burnout point to an opponent's employee.

---

*Have you tried turning it off and on again?*

## Downer Dave

1

2

3



Employee - DEV

1

### When played:

add (1) burnout point to an opponent's employee.

---

*Have you tried turning it off and on again?*

# Database Developer

1 2 3 4



## When played:

draw (2) cards. For any action card drawn this way:  
you can play it this turn for (0).

---

*Drop database? I meant drop the bass!*

# Database Developer

1 2 3 4



## When played:

draw (2) cards. For any action card drawn this way:  
you can play it this turn for (0).

---

*Drop database? I meant drop the bass!*

# BI Developer

1 2 3



## When played:

draw a card. If the drawn card is a knowledge card:  
you can play it this turn for (2) less.

---

*Let's go to the Foo Bar!*

# BI Developer

1 2 3



## When played:

draw a card. If the drawn card is a knowledge card:  
you can play it this turn for (2) less.

---

*Let's go to the Foo Bar!*

# Full Stack Ninja

1 2 3 4



Employee - DEV

3

Playing and transferring knowledge cards on this developer costs (2) less.

---

*I know HTML, CSS and some Minecraft Development.*

# Full Stack Ninja

1 2 3 4



Employee - DEV

3

Playing and transferring knowledge cards on this developer costs (2) less.

---

*I know HTML, CSS and some Minecraft Development.*

# QA Wizard Engineer

1 2 3



Employee - DEV

2

## While in play:

your developers cost (1) less.

---

*Deleting cookies is an important part of my job.*

# QA Wizard Engineer

1 2 3



Employee - DEV

2

## While in play:

your developers cost (1) less.

---

*Deleting cookies is an important part of my job.*

# Senior Backend Developer

1

2

3

4



Employee - DEV

2

## When played:

look at the top (5) cards of the deck and draw (2) of them. Put the rest on the bottom of the deck.

---

*Deploy on Friday, they said. It'll be fun, they said.*

# Senior Backend Developer

1

2

3

4



## When played:

look at the top (5) cards of the deck and draw (2) of them. Put the rest on the bottom of the deck.

---

*Deploy on Friday, they said. It'll be fun, they said.*

# Junior Backend Developer

1

2



## When played:

look at the top (3) cards of the deck and draw one of them. Put the rest on the bottom of the deck.

---

*Normally I miss deadlines like a stormtrooper misses Jedi.*

# Junior Backend Developer

1

2



## When played:

look at the top (3) cards of the deck and draw one of them. Put the rest on the bottom of the deck.

---

*Normally I miss deadlines like a stormtrooper misses Jedi.*

# Research Engineer

1

2



## When played:

look at the top (3) cards of the deck and put them back in any order.

---

*Research is what I'm doing when I don't know what I'm doing.*

# Research Engineer

1

2



Employee - DEV

2

## When played:

look at the top (3) cards of the deck and put them back in any order.

---

*Research is what I'm doing when I don't know what I'm doing.*

## Lead Developer

1 2 3 4 5



### While in play:

transferring knowledge between developers costs you (2) less.

---

*In order to understand recursion, one must first understand recursion.*

## Lead Developer

1 2 3 4 5



### While in play:

transferring knowledge between developers costs you (2) less.

---

*In order to understand recursion, one must first understand recursion.*

Intern

1



Employee - DEV

1

...

---

*I have no idea what I'm doing.*

Intern

1



Employee - DEV

1

...

---

*I have no idea what I'm doing.*

# Shy Frontend Dev

1

2



Employee - DEV

1

**Pay (x):**

this developer gains (x) efficiency until the end of your turn.

---

*Working overtime again?*

## Office Manager

1 2 3



Employee - HR



### When played:

remove (1) burnout point from all your employees.

**At the end of your turn:** draw a card.

---

*Entry level positions require 2 years of experience?*

## Office Manager

1 2 3



Employee - HR



### When played:

remove (1) burnout point from all your employees.

**At the end of your turn:** draw a card.

---

*Entry level positions require 2 years of experience?*

## Rockstar Recruiter

1

2

3



### When played:

if an opponent has (4) or more employees: take over a target employee.

**At the end of your turn:** draw a card.

---

*I'm not stalking you. It's social research.*

## Rockstar Recruiter

1

2

3



Employee - HR



### When played:

if an opponent has (4) or more employees: take over a target employee.

**At the end of your turn:** draw a card.

---

*I'm not stalking you. It's social research.*

## Undercover HR Agent

1

2

3



Employee - HR



### Once a turn you may pay (1):

declare a coin flip. If you win: **take over a target employee.** **Before the coin flip** any opponent may pay you (3) to counter this ability.

---

*Our new employee used to write spaghetti code for a big pasta company.*

## Undercover HR Agent

1

2

3



Employee - HR



### Once a turn you may pay (1):

declare a coin flip. If you win: **take over a target employee.** **Before the coin flip** any opponent may pay you (3) to counter this ability.

---

*Our new employee used to write spaghetti code for a big pasta company.*

# The Nice HR Lady

1

2



Employee - HR



**At the end of your turn:** draw a card.

---

*It's nice to be important, but it's important to always be nice.*

## The Nice HR Guy

1

2



Employee - HR



**At the end of your turn:** draw a card.

---

*I love it when someone insults me. That means that I don't have to be nice anymore.*

36/64

## Scrum Master

1

2

3

4



Employee - Agile Coach

+1

### When played:

draw a card for each developer you control.

**While in play:** your developers have efficiency +1.

---

*If you can't convince them, confuse them.*

## Scrum Master

1 2 3 4



Employee - Agile Coach

+1

### When played:

draw a card for each developer you control.

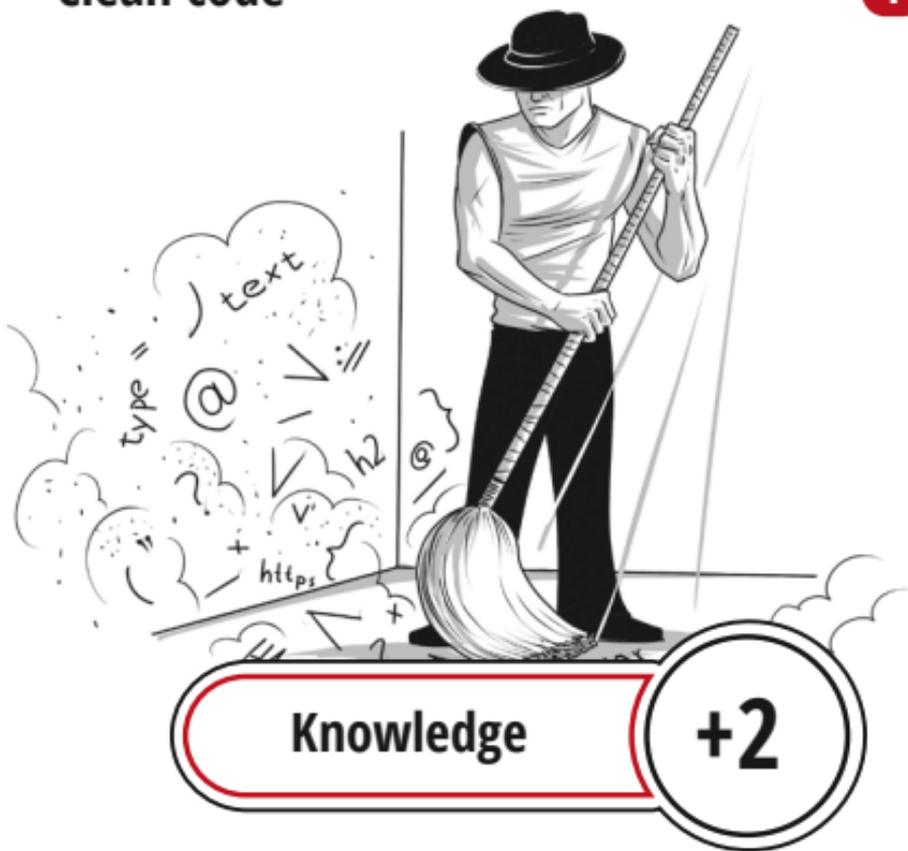
**While in play:** your developers have efficiency +1.

---

*If you can't convince them, confuse them.*

# Clean code

1



Attach this card to one of your developers to improve their efficiency. Knowledge can be transferred to another developer by paying its cost.

*It is not enough for code to work.*

## Design patterns

1

2



Knowledge

+3

Attach this card to one of your developers to improve their efficiency. Knowledge can be transferred to another developer by paying its cost.

---

*A design that doesn't take change into account risks major redesign in the future.*

## Defensive programming

1

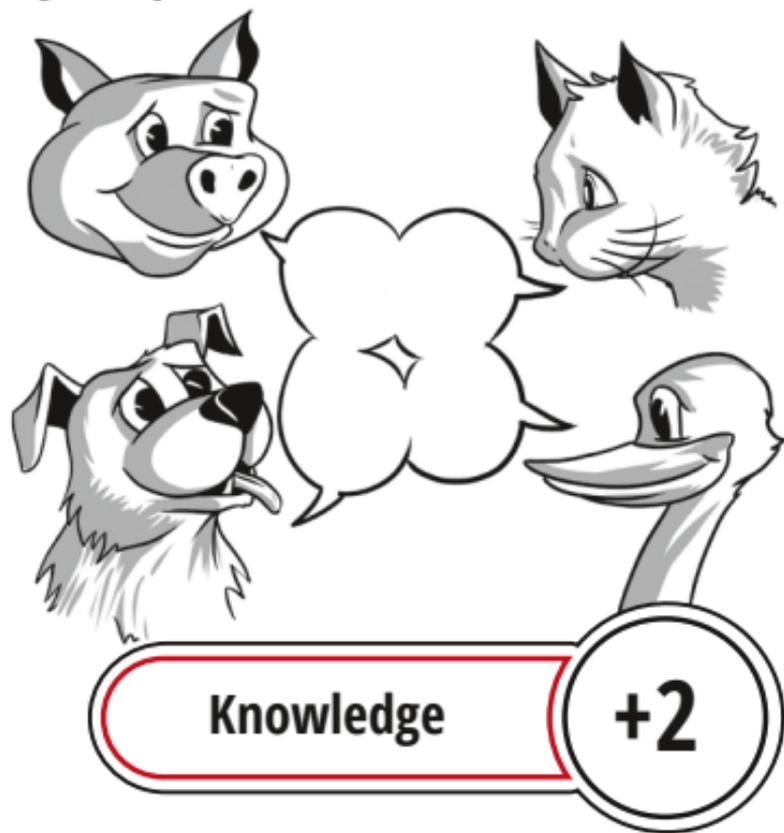


Attach this card to one of your developers to improve their efficiency. Knowledge can be transferred to another developer by paying its cost.

*The only defense against the world is a thorough knowledge of it.*

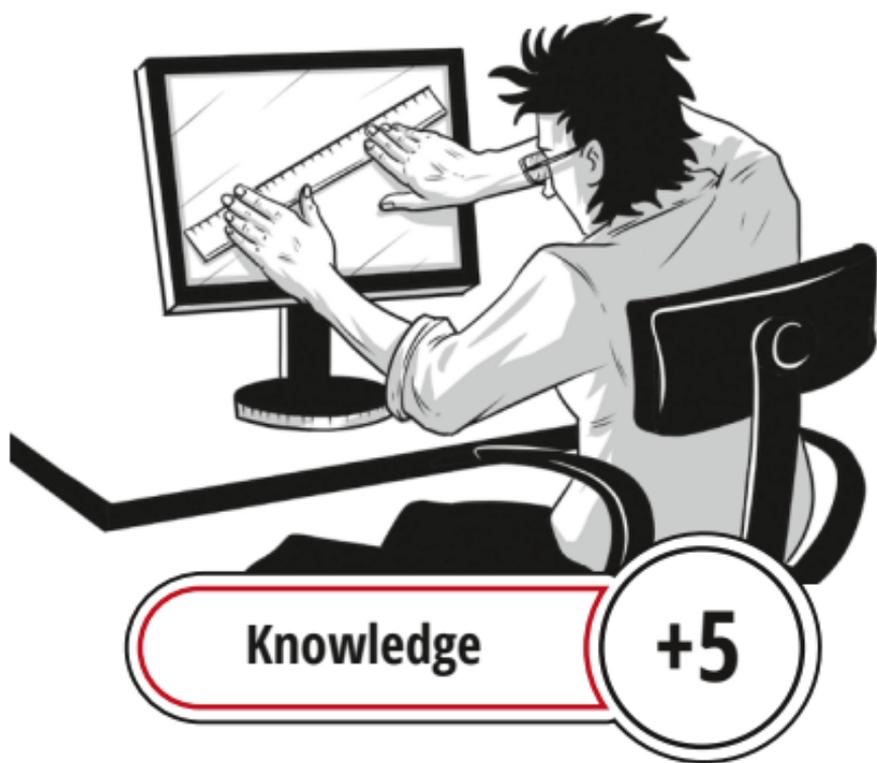
# Polymorphism

1



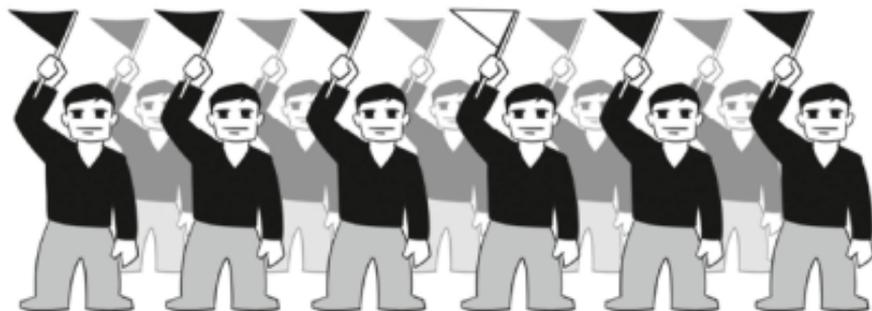
Attach this card to one of your developers to improve their efficiency. Knowledge can be transferred to another developer by paying its cost.

*Inheritance is the object-oriented way to become wealthy.*



Attach this card to one of your developers to improve their efficiency. Knowledge can be transferred to another developer by paying its cost.

*Test first, debug later.*



**Knowledge**

**+4**

Attach this card to one of your developers to improve their efficiency. Knowledge can be transferred to another developer by paying its cost.

*If you don't know the purpose of the test, you are free to delete it.*

**SOLID**

1

2



**Knowledge**

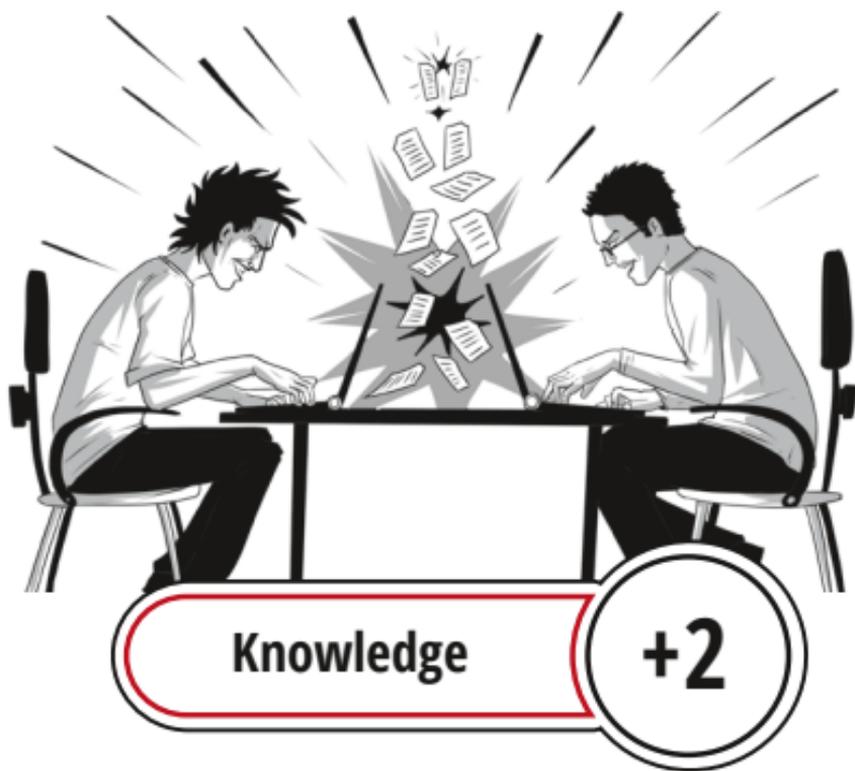
**+3**

Attach this card to one of your developers to improve their efficiency. Knowledge can be transferred to another developer by paying its cost.

*Knowledge is power.*

## Version control

1



Knowledge

+2

Attach this card to one of your developers to improve their efficiency. Knowledge can be transferred to another developer by paying its cost.

*In case of fire: commit, push and leave the building!*

## Domain knowledge

1

2

3



Attach this card to one of your developers to improve their efficiency. Knowledge can be transferred to another developer by paying its cost.

---

*If you're the smartest person in the room, you're in the wrong room.*

## Dependency injection

1

2



**Knowledge**

**+3**

Attach this card to one of your developers to improve their efficiency. Knowledge can be transferred to another developer by paying its cost.

*Now we can write sweet tests for our Cookie class!*

# Debugging

1

2



Knowledge

+3

Attach this card to one of your developers to improve their efficiency. Knowledge can be transferred to another developer by paying its cost.

*Some bugs in our backlog transformed into awesome features!*

## Continuous integration

1

2

3



Attach this card to one of your developers to improve their efficiency. Knowledge can be transferred to another developer by paying its cost.

*Today we will introduce you to MR. Jenkins!*

# Outsourcing

1

2



## Action Card



Take over a target employee until the start of your next turn. The opponent gets the resources you paid for this card.

*I used to outsource all my homework to someone from another country.*

# Outsourcing

1

2



## Action Card



Take over a target employee until the start of your next turn. The opponent gets the resources you paid for this card.

*I used to outsource all my homework to someone from another country.*

# Investor

0



You get (2) additional resources.

Playing this card costs (0).

---

*Sometimes all you need is a million dollars.*

# Investor

0



You get (2) additional resources.

Playing this card costs (0).

---

*Sometimes all you need is a million dollars.*

# Investor

0



You get (2) additional resources.

Playing this card costs (0).

---

*Sometimes all you need is a million dollars.*

# Investor

0



You get (2) additional resources.

Playing this card costs (0).

---

*Sometimes all you need is a million dollars.*

# Head Hunter

1 2 3 4 5



Action Card



Take over a target employee. The opponent can pay you (5) to counter this card.

---

*Don't sell yourself short, my friend.*

## Coffee machine

1

2



You get (1) project point for every developer you control. All your employees lose (1) burnout point.

---

*Let's turn caffeine into code!*

## Get away from it all

1

2

3



### Action Card



All your employees lose (1) burnout point and they don't get any burnout points at the end of this turn.

---

*I need a six month vacation, twice a year.*

# Get away from it all

1 2 3



## Action Card



All your employees lose (1) burnout point and they don't get any burnout points at the end of this turn.

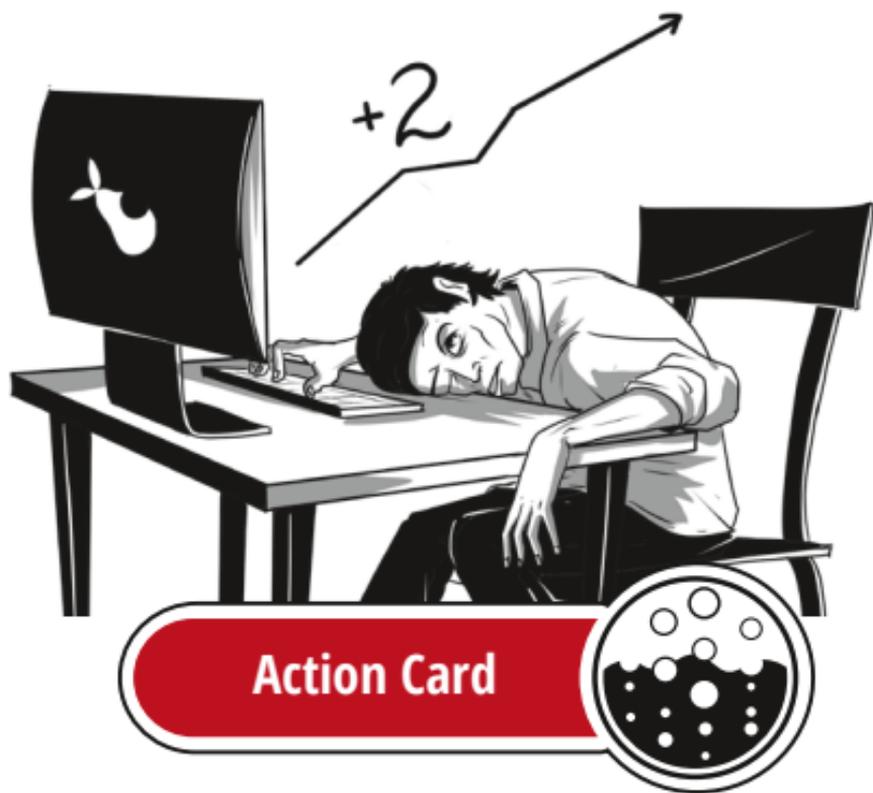
---

*I need a six month vacation, twice a year.*

## Crunch time

1

2



### Action Card

You get (2) project points for every developer you control. At the end of your turn every developer you control gets (1) burnout point.

*Work-life balance? What's that?*

## Technical debt

1 2 3 4



Shuffle all employees and their knowledge into the deck.

---

*Are you brave enough to touch the ancient legacy code?*

# Monster Bug

1

2

3



Action Card



All opponent employees get (1) burnout point.

---

*It's not a bug, it's a feature.*

# Monster Bug

1 2 3



Action Card



All opponent employees get (1) burnout point.

---

*It's not a bug, it's a feature.*