



## Regolamento

**Commit & Conquer** è un gioco di dadi in cui due o più developer si scontrano per portare a termine un progetto il più velocemente possibile, con la qualità più alta possibile.

Per completare il progetto i developer dovranno fare dei *commit*, aprire *pull request* e chiedere ai propri colleghi di fare *code review*. Potranno anche consultare la nota piattaforma *HeapOverrun*, da cui copiare codice di dubbia provenienza, che velocizzerà il lavoro aumentando il rischio di *bug*.

Scopo del gioco è arrivare a **20 punti** prima degli altri developer.

Come prima cosa, ogni developer sceglie il linguaggio in cui è senior:

Javascript (1) | Python (2) | Java (3) | C++ (4) | PHP (5) | Ruby (6)

Il gioco si svolge a turni, durante i quali un developer può fare una delle seguenti azioni:

- **commit**: aggiungi 1 dado alla riserva
- **copiare da HeapOverrun**: aggiungi 2 dadi alla riserva, ignora il linguaggio senior
- **leggere la documentazione**: ritiri un dado al *push*
- **chiedere una code review**: elimini un *bug* al *push*
- **push**: lanci tutti i dadi accumulati finora nella riserva

## Accumulare punti

Una volta fatto il *push*, il developer svuota la riserva e guadagna punti:

- **+1 punto** per il *commit* iniziale
- **+2 punti** per ogni *commit* oltre quello iniziale
- **-3 punti** per ogni *bug* ottenuto

## Bug

Il developer ottiene un bug per ogni dado che ottiene lo stesso risultato di un altro dado:

```
esempioBug = 5 dadi lanciati => 2, 3, 3, 5, 6 => ci sono 2 bug (3, 3)
```

Il developer ignora il primo dado che ottiene il punteggio corrispondente al suo linguaggio senior, quando calcola i bug:

```
esempioBugSenior_1 = 5 dadi lanciati => 3, 3, 3, 5, 6 => ci sono 2 bug (3, 3)
```

```
esempioBugSenior_2 = 5 dadi lanciati => 2, 3, 3, 5, 6 => non ci sono bug
```

## Riserva

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## Linguaggio Senior

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6

## HeapOverrun

☐

## Code Review

☐ ☐ ☐

## Documentazione

☐ ☐ ☐



# SPARKFABRIK

[www.sparkfabrik.com](http://www.sparkfabrik.com)

**SparkFabrik** progetta e sviluppa applicazioni **Cloud Native** attraverso le più recenti tecniche **DevSecOps**, con un occhio alla **Developer Experience** interna e per i suoi clienti.

Supporta le aziende nei complessi processi di digital transformation con il suo **Cloud Native Journey**, un modello di adozione nato da ricerca ed esperienza concreta sul campo.

E siamo gente che ci crede e si diverte. Ascoltateci, leggeteci e giocate con noi!



**Spark e la CNCF**

[bit.ly/Spark-CNCF](http://bit.ly/Spark-CNCF)



**GitHub Spark**

[bit.ly/Spark-GH](http://bit.ly/Spark-GH)



**Il nostro  
Playbook**

[bit.ly/Spark-PB](http://bit.ly/Spark-PB)



**Continuous  
Delivery**

[bit.ly/Spark-CD](http://bit.ly/Spark-CD)