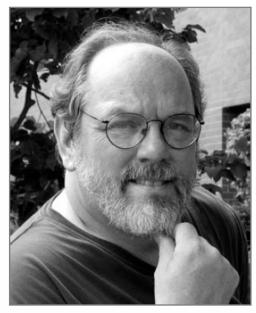
Tech Debt

YEGOR BUGAYENKO

Lecture #14 out of 24 80 minutes

The slidedeck was presented by the author in this YouTube Video

All visual and text materials presented in this slidedeck are either originally made by the author or taken from public Internet sources, such as web sites. Copyright belongs to their respected authors.



WARD CUNNINGHAM

"Shipping first time code is like going into <u>debt</u>. A little debt speeds development so long as it is paid back promptly with a rewrite. The danger occurs when the debt is not repaid. Every minute spent on not-quite-right code counts as interest on that debt."

— Ward Cunningham. Experience Report — the WyCash Portfolio Management System. In *Proceedings of the Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*, pages 29–30, 1992. doi:10.1145/157710.157715

Puzzle Driven Development: Motivating Example

Commit #1:

```
int fibonacci(int n) {
  if (n <= 2) {
    return 1;
  }

// @todo I don't know

// what to do when "n"

// is larger than "2".

// Implement it and uncomment

// the assertion below.

return 0;

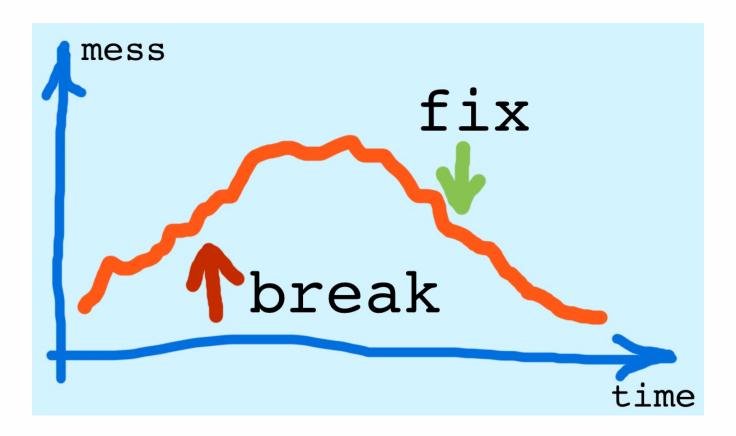
sassert fibonacci(0) == 1;
assert fibonacci(2) == 1;
// assert fibonacci(9) == 34;</pre>
```

Commit #2:

```
int fibonacci(int n) {
   if (n <= 2) {
     return 1;
   }
   if (n == 9) {
     return 34;
   }
   // @todo Implement others
   // too, but I don't know
   // how to do it right.
   return 0;
}
assert fibonacci(2) == 1;
assert fibonacci(9) == 34;
// assert fibonacci(10) == 55;</pre>
```

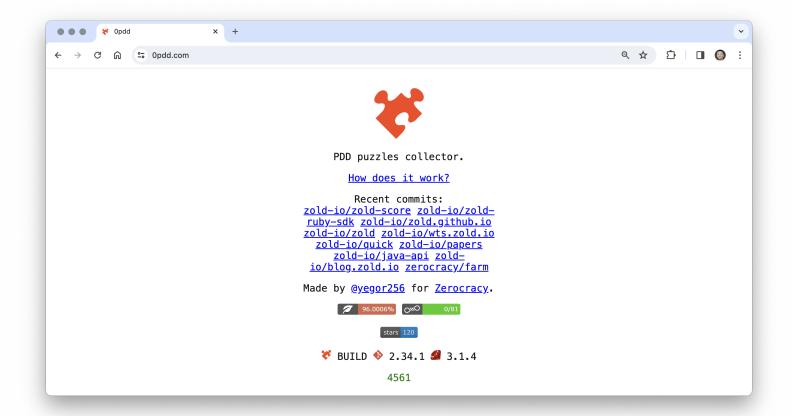
```
16 \end{ffcode*}
17 | }
18 \par\columnbreak\par
19 | Commit \#3:\par
20 {\scriptsize\begin{ffcode}
21 int fibonacci(int n) {
   if (n \le 2) {
   return 1;
24
   return fibonacci(n-1)
      + fibonacci(n-2);
27 | }
28 assert fibonacci(0) == 1;
29 assert fibonacci(2) == 1;
30 assert fibonacci(9) == 34;
31 assert fibonacci(10) == 55;
```

Break-and-Fix Cycle

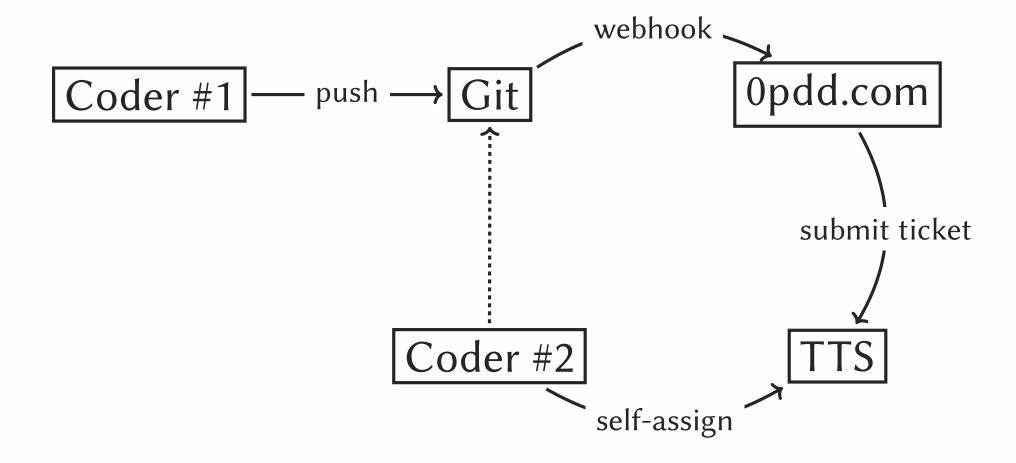


Source: https://www.yegor256.com/2014/04/12/puzzle-driven-development-by-roles.html

www.Opdd.com



PDD Pipeline



250+ Puzzles in objectionary/eo

```
** objectionary/eo x +

** Opd. := Opdd.com/p?name=objectionary/eo

** Opd. := Opdd.com/p.name=objectionary/eo

** Opdd.com/p.name=objectionary/eo

** Opdd.com/p.name=objectionary/eo

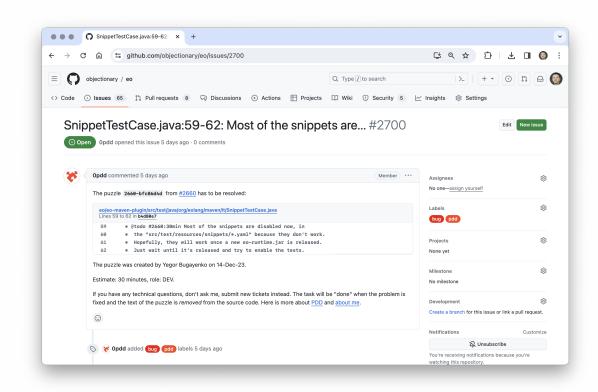
** Opdd.com/p.name=objectionary/eo

** Opdd.com/p.name=objectionary/eo

** Opdd.com/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=objectionary/eo/p.name=obje
```

Source: https://www.Opdd.com/p?name=objectionary/eo

Sample Ticket Submitted by Opdd.com to objectionary/eo



Source: https://github.com/objectionary/eo/issues/2700



YEGOR BUGAYENKO, MIRKO FARINA, ARTEM KRUGLOV, WITOLD PEDRYCZ, YAROSLAV PLAKSIN, GIANCARLO SUCCI

"This paper presents the benefits of considering the entire backlog when prioritizing tasks. We employ an iterative approach using Particle Swarm Optimization to optimize a linear model with various preprocessing methods to determine the optimal model for task prioritization within a backlog."

— Yegor Bugayenko, Mirko Farina, Artem Kruglov, Witold Pedrycz, Yaroslav Plaksin, and Giancarlo Succi. Automatically Prioritizing Tasks in Software Development. *IEEE Access*, 2023. doi:10.1109/access.2023.3305249

Read this:

Puzzle Driven Development (2010)

PDD by Roles (2014)

PDD in Action (2017)

References

Yegor Bugayenko, Mirko Farina, Artem Kruglov, Witold Pedrycz, Yaroslav Plaksin, and Giancarlo Succi. Automatically Prioritizing Tasks in Software Development. *IEEE Access*, 2023. doi:10.1109/access.2023.3305249.

Ward Cunningham. Experience Report — the WyCash Portfolio Management System. In Proceedings of the Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), pages 29–30, 1992. doi:10.1145/157710.157715.