

Ensuring quality for a peculiar piece of software



2 October 2025

Joep Schuurkes

Staff Test Engineer
Dutch Electoral Council

[@joeposaurus@chaos.social](mailto:joeposaurus@chaos.social)
<https://smallsheds.garden>

What's peculiar about it?

used once a year (or not at all, or multiple times)

350+ organisations

most users are volunteers

delivery weeks before usage

air-gapped server and clients

Why is it peculiar?

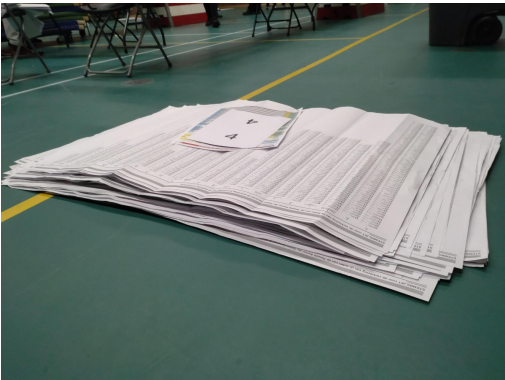
software for Dutch elections

voting on paper ballots

data entry per polling station

adding up all the votes

determining who gets a seat



One way to build it

clear requirements

clear deadline

very few and stable integrations

lock ourselves away for 18 months

Another way to build it

modern software development practices still apply

always be delivering

always be collecting feedback

The four practices I will cover

use cases

thin vertical slices

zero bug policy

working in public

quality is many small things
done right

Not just those four, also

just-in-time refinement

domain knowledge

purpose

pipeline with quality gates

and many more...

My role in all of this

test & quality engineer

not testing every change
attending to quality in general
which includes some testing

use cases

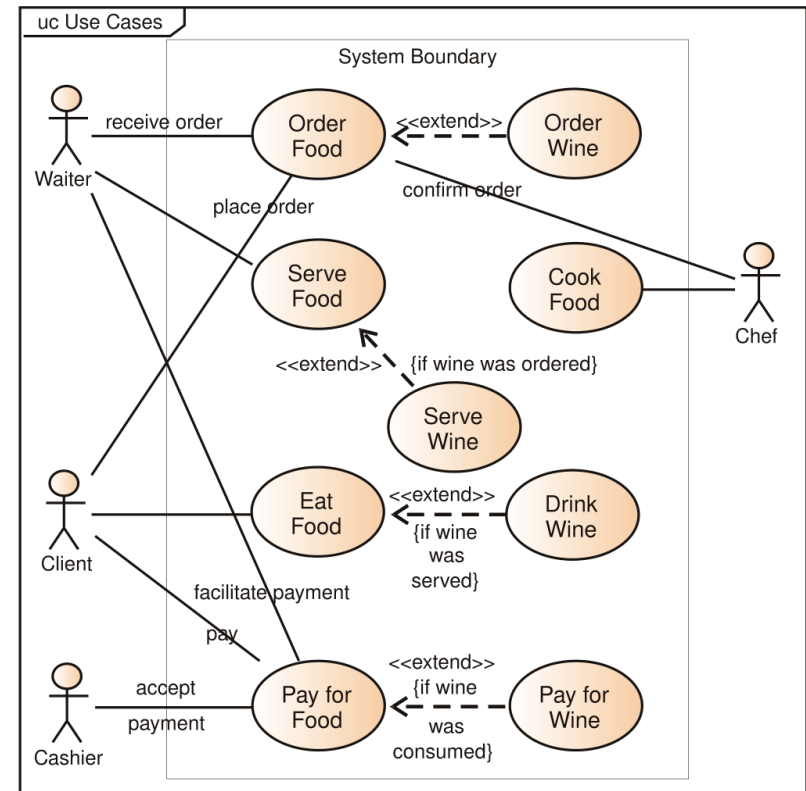
My previous experiences

thick documents

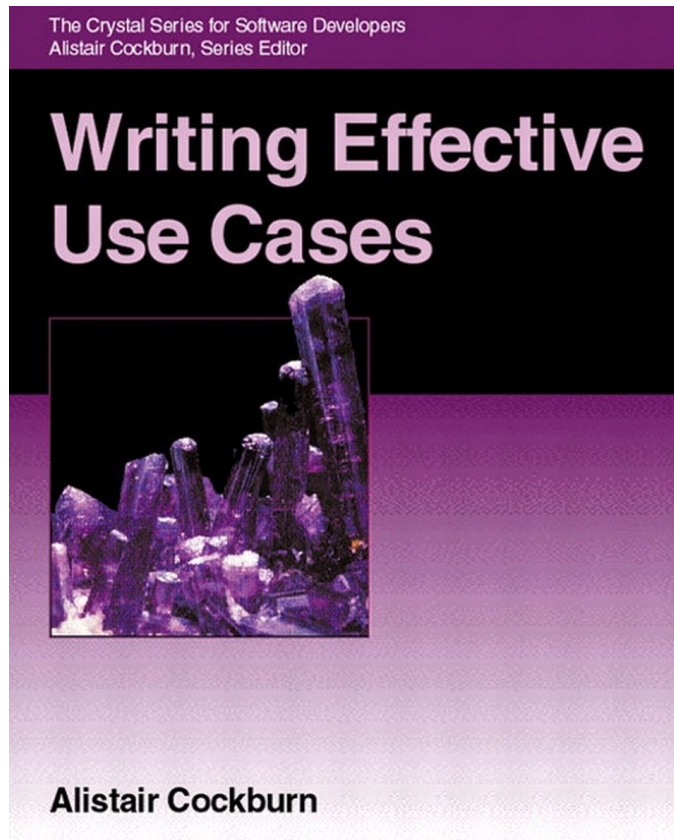
confusing diagrams

too much detail

done before I start



There's a book!



Writing Effective Use Cases
– Alistair Cockburn

Why not user stories?

user stories are for work management



Tree structure

 cloud

 kite

 sea

 fish

 shell

Effective use cases

subject – verb – object

"The application determines the data is valid."

different levels of formality

Use case example

De beheerder richt de applicatie in voor GSB en/of CSB (wolk)

Niveau: hoog-over, wolk, ☁

Hoofdscenario en uitbreidingen

1. [De beheerder installeert de applicatie.](#)
2. De beheerder maakt voor zichzelf een eerste account en wachtwoord aan.
3. [De beheerder zet de verkiezingen in de applicatie.](#)
4. De beheerder leest de kandidatenlijst in.
5. [De beheerder leest het bestand met stembureaus en aantal kiesgerechtigden in.](#)
6. De beheerder selecteert het type stemopneming.
7. De beheerder bevestigt het aantal kiesgerechtigden in de gemeente.
8. [De beheerder maakt de gebruikers aan.](#)

Uitbreidingen:

- 4a. De applicatie geeft een foutmelding bij het inlezen van de kandidatenlijst:
- 5a. De beheerder slaat het invoeren van stembureaus over:
 - 5a1. (na inrichten applicatie) [De beheerder of coördinator zet de stembureaus in de applicatie.](#)

How use cases have helped us

better understanding by team

better understanding by stakeholders

input for UX design

input for epic kick-off

double check before closing epic

The reality

requires investing energy

some people engage more than others

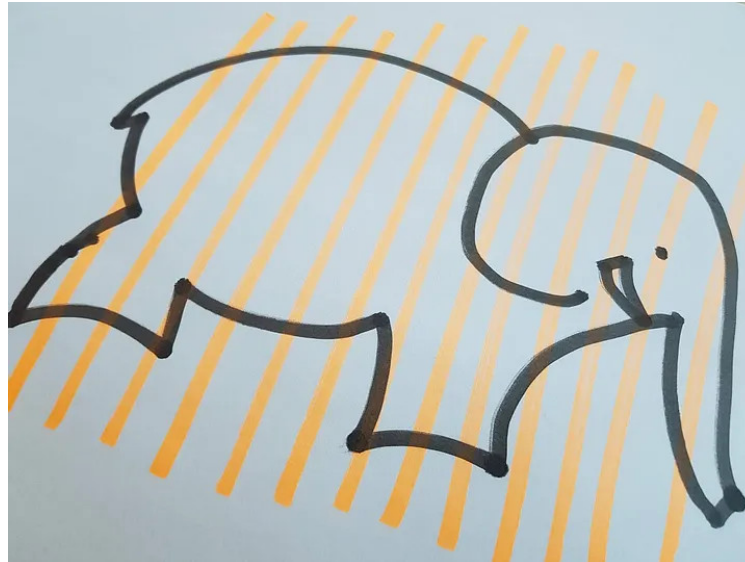
short-term and long-term value

thin vertical slices

Slices, thin and vertical

how do you split up the work?

Elephant Carpaccio-workshop



Thin slices

small steps of progress

instead of...

feedback on what you're working on now

instead of last week

Vertical slices

frontend – backend – database

instead of...

rich feedback

instead of narrow feedback

The reality

thin-ish and vertical-ish slices

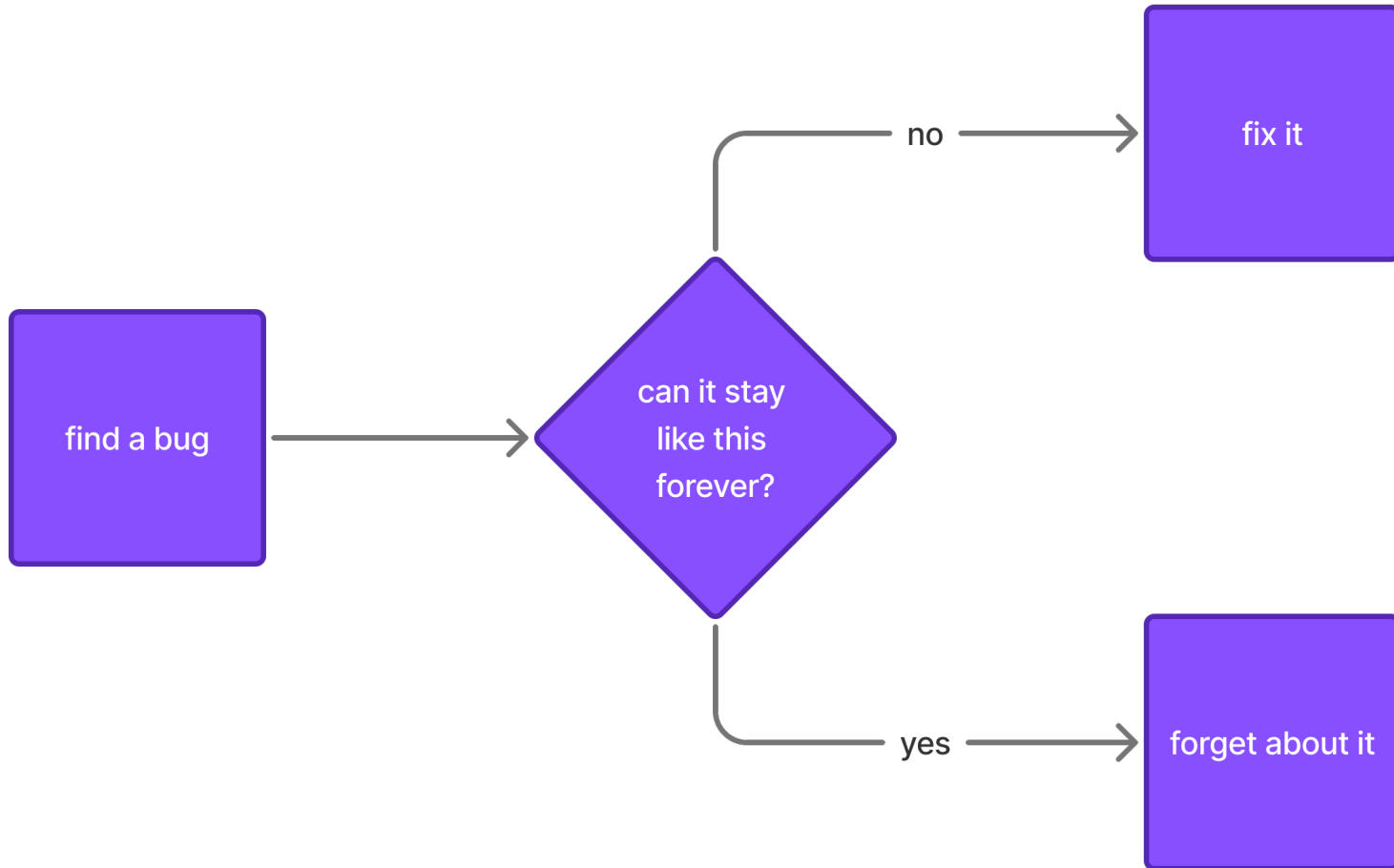
small epics

challenge of incremental feedback

challenge of greenfield project

zero bug policy

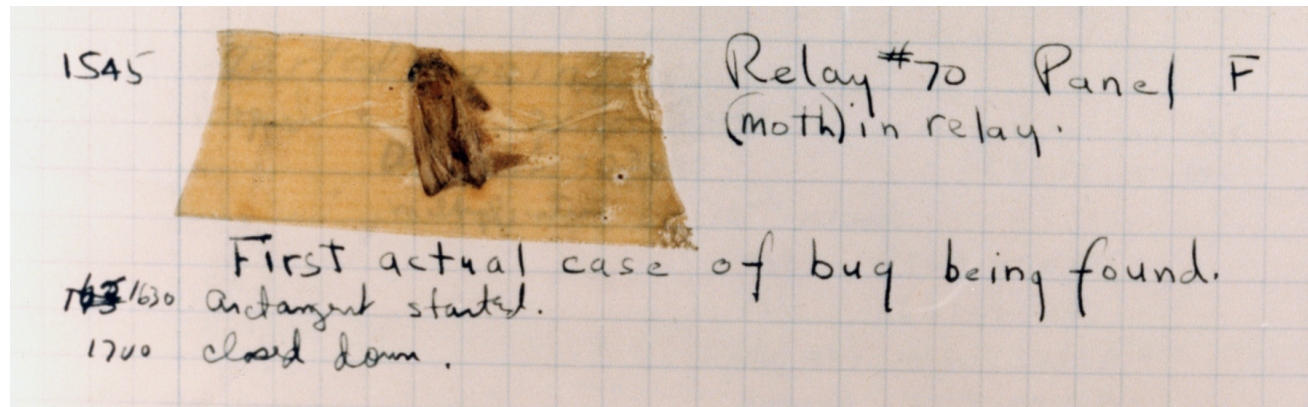
Zero bug policy



The why

no backlog full of bugs

done is done,
so done needs to be good enough



The reality

discipline and vigilance

team culture

recently postponed a fix

working in public

It's all public!

the code

the repository github.com/kiesraad/abacus

the designs [figma](#)

the pipelines and releases

the documentation

the project board

the test environments abacus-test.nl

But why?

because we have to

because we think we should



What it has brought us so far

fuzz testing

June 10, 2025

Putting seat calculations in Dutch election software to the (fuzz) test



Marc
Systems software engineer




Michiel
Software engineer

development

open-source

testing

 rust

What it has brought us so far

six external pull requests (so far)

Small documentation correction: missing a seat in test text of full seats for this test case #1651

[Preview](#) [Switch back](#) [Feedback](#) [Edit](#) [Code](#)

Merged [praseodym](#) merged 1 commit into [kiesraad:main](#) from [Hilbrand:patch-1](#) on Jun 24

Conversation 2 Commits 1 Checks 16 Files changed 1

+1 -1

All commits

0 / 1 viewed

Comments 0

Filter files...

backend/src/apportionment

seat_assignment.rs

backend/src/apportionment/seat_assignment.rs

+1 -1

Viewed

```
@@ -1114,7 +1114,7 @@ mod tests {
```

```
1114 1114
```

```
1115 1115     /// Apportionment with residual seats assigned with largest remainders method
```

```
1116 1116     ///
```

```
1117 -     /// Full seats: [6, 3, 3, 0, 0, 0, 0] - Remainder seats: 3
```

```
1117 +     /// Full seats: [6, 3, 3, 0, 0, 0, 0, 0] - Remainder seats: 3
```

```
1118 1118     /// Remainders: [0/15, 0/15, 0/15, 55, 50, 45, 45, 45], only votes of lists [1, 2, 3] meet the threshold of 75% of the quota
```


```
1119 1119     /// 1 - largest remainder: seat assigned to list 1
```

```
1120 1120     /// 2 - largest remainder: seat assigned to list 2
```

What it has brought us so far


involvement of municipalities

41 minutes ago

 github-actions

 v0.1.0-alpha....

 fdaebeef 











Compare 

Version v0.1.0-alpha.20250921.fdaebeef

Pre-release

This is an alpha release of Abacus, commit [fdaebeef](#) on 2025-09-21

Assets 6

 abacus-linux-v0.1.0-alpha.20250921.fdaebeef	sha256:97d242436029fb...		62.1 MB	41 minutes ago
 abacus-macos-v0.1.0-alpha.20250921.fdaebeef	sha256:abe0f032de2db4...		55.9 MB	41 minutes ago
 abacus-windows-v0.1.0-alpha.20250921.fdaebeef.exe	sha256:bb165f1ffa4aba...		55.4 MB	41 minutes ago
 SHA256SUMS	sha256:36f8e4f5fdb621...		333 Bytes	41 minutes ago
 Source code (zip)				2 days ago
 Source code (tar.gz)				2 days ago



The reality

it takes more than making the repository public

documentation

reaching out to people

responding to feedback

The four practices I covered

use cases

thin vertical slices

zero bug policy

working in public

closing thoughts

this basic Agile stuff(TM),
it works!

adapt practices based on
outcome(s) and principles

We Tried Baseball and It Didn't Work – Ron Jeffries

quality is something emergent

- Anne-Marie Charrett

you don't test quality in

you don't sprinkle on
a little quality at every step

quality is many small things
done right

thank you!



slides at smallsheds.garden/my-talks/