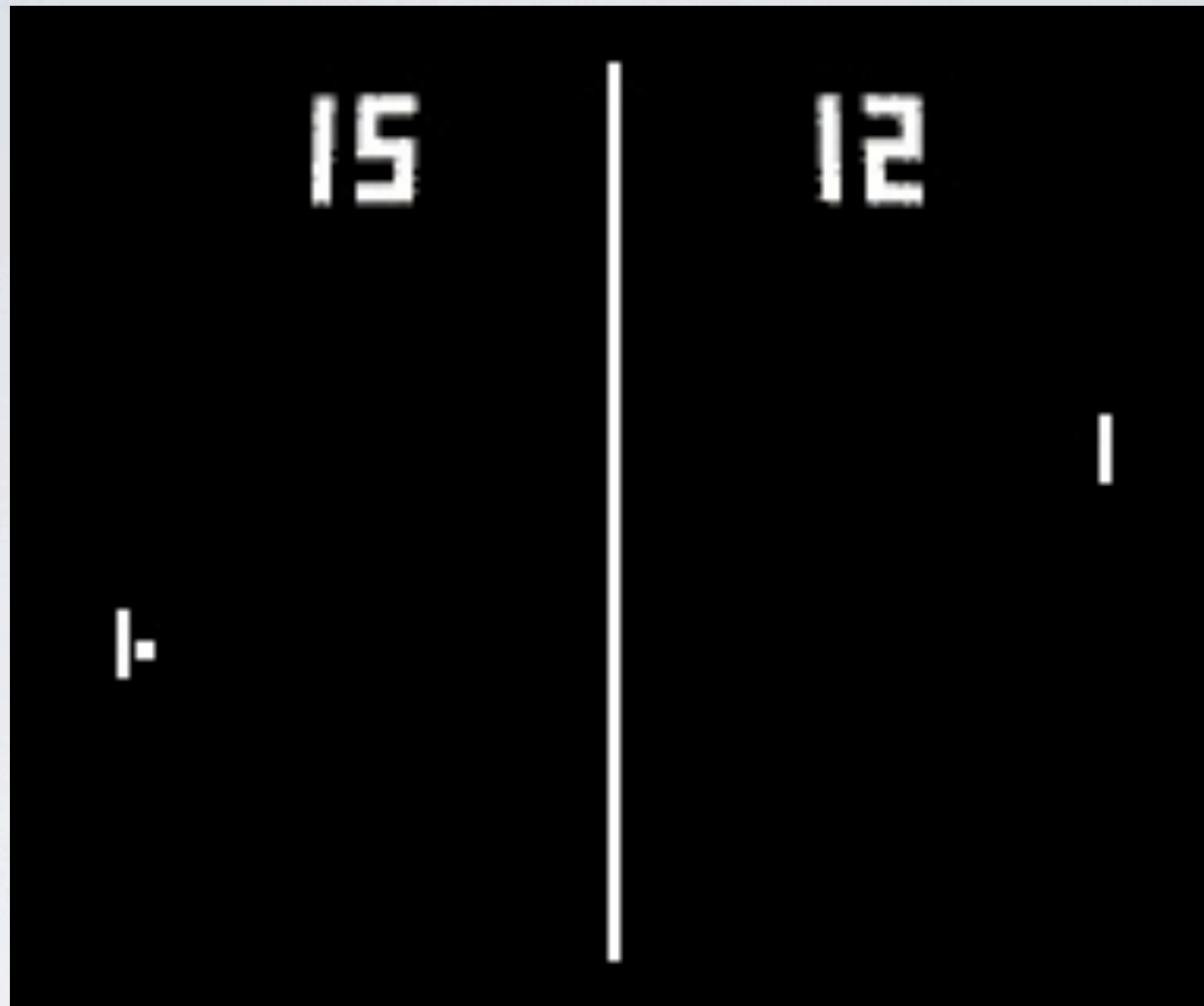


THE NEXT ITERATION OF YOUR GUI

PEP 342 (Coroutines via Enhanced Generators)
PEP 380 (Syntax for Delegating to a Subgenerator)

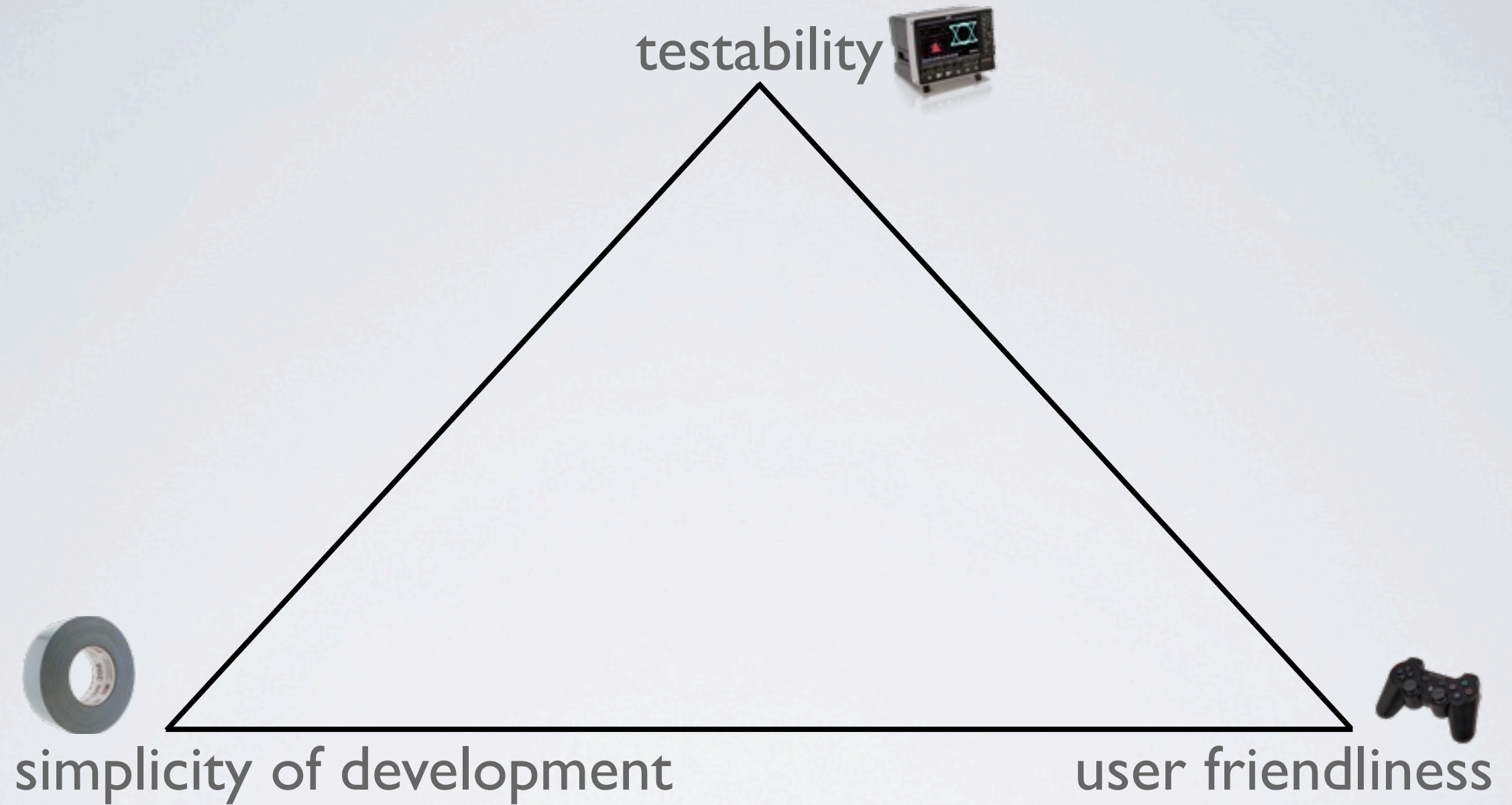


conceptive.be @erikjanss
shore.be @jeroendierckx



WHAT'S GOING ON

interaction



TRADEOFFS



TRADEOFFS

user friendliness

- use a progress indicator
- provide a cancel button
- respond to events
- similar controls as in other places



USER FRIENDLINESS

OSX human interface guidelines



USER FRIENDLINESS



Multithreaded toolkits - A failed dream?

Graham Hamilton

https://weblogs.java.net/blog/kgh/archive/2004/10/multithreaded_t.html



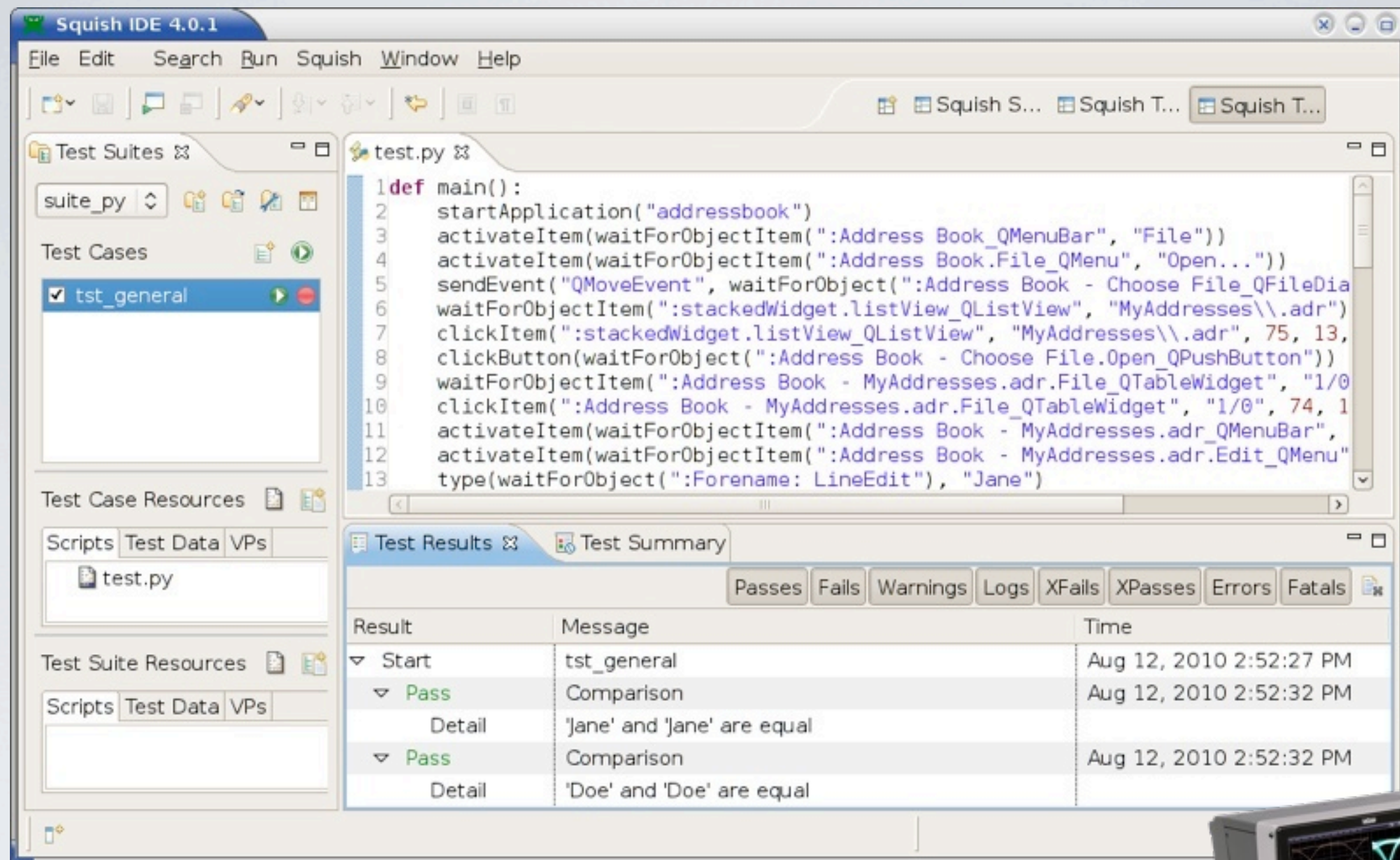
TRADEOFFS

simplicity of development



TRADEOFFS

testability



TRADEOFFS

testability



EXAMPLE APPLICATION

process pickinglist



example application

PROCESS PICKINGLIST

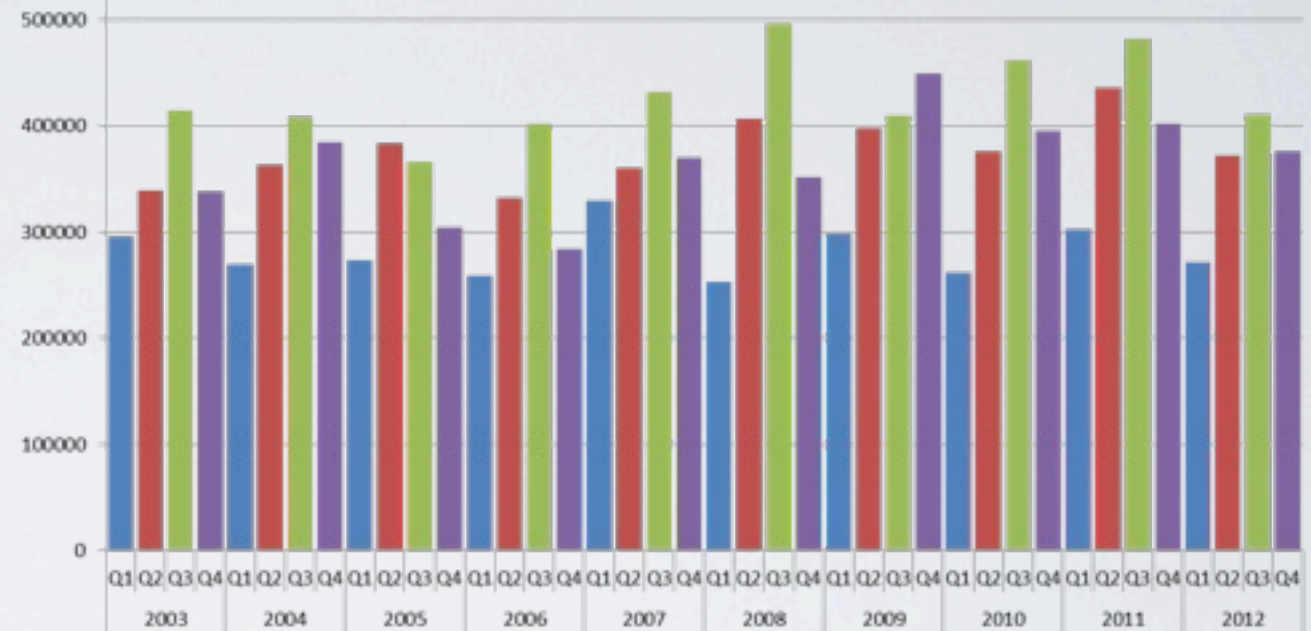
step one. scan items in warehouse



example application

PROCESS PICKINGLIST

step two. import scanned lines



example application

PROCESS PICKINGLIST

step two. import scanned lines

Order - Scans

Data Preview

Please review the data below.

	Artikelnummer	Omschrijving	Gescand	Besteld	Geannuleerd	Geleverd	Klantnummer	statu
	3029	Skipping rope 2 n	0	25	0	0	108001	FOR
	3516	Skipping rope 2 n	3	25	0	0	108001	FOR
	3028	Skipping rope 2 n	1	25	0	0	108001	FOR
	3514	Skipping rope 2 n	0	25	0	0	108001	FOR
	3581	Jump ropes 10 pic	0	24	0	0	108001	FOR
	3584	Plaited rope 10 m	8	4	0	0	108001	FOR
	36238	Soft Bowling	0	12	0	0	108001	FOR
	33507	Set Cones with N	12	10	0	0	108001	FOR
	35439	Pitch the Rings	0	12	0	0	108001	FOR

Cancel OK

example application

PROCESS PICKINGLIST

step three. verify import data



example application

PROCESS PICKINGLIST

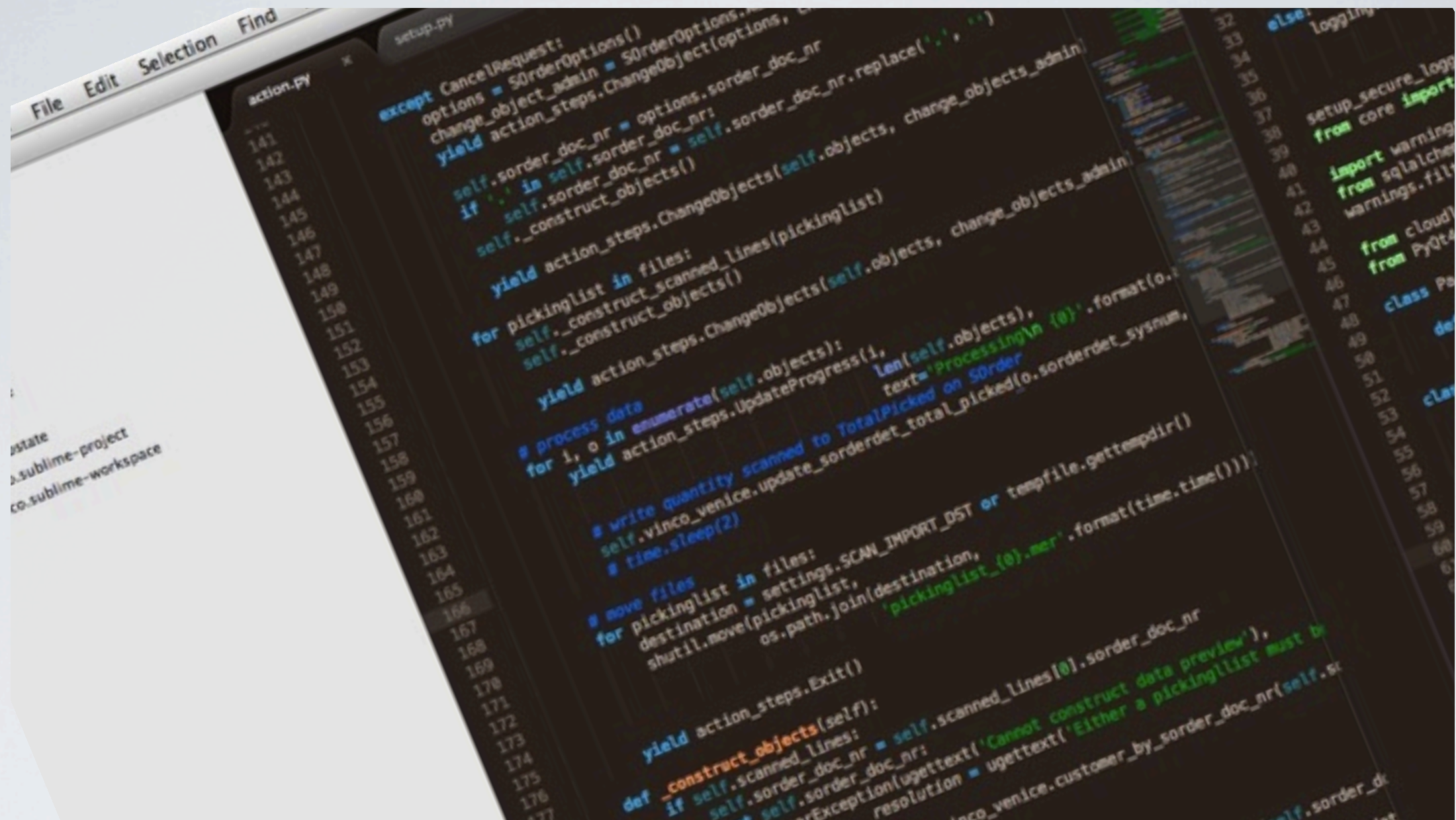
possible issues. freeze



example application

PROCESS PICKINGLIST

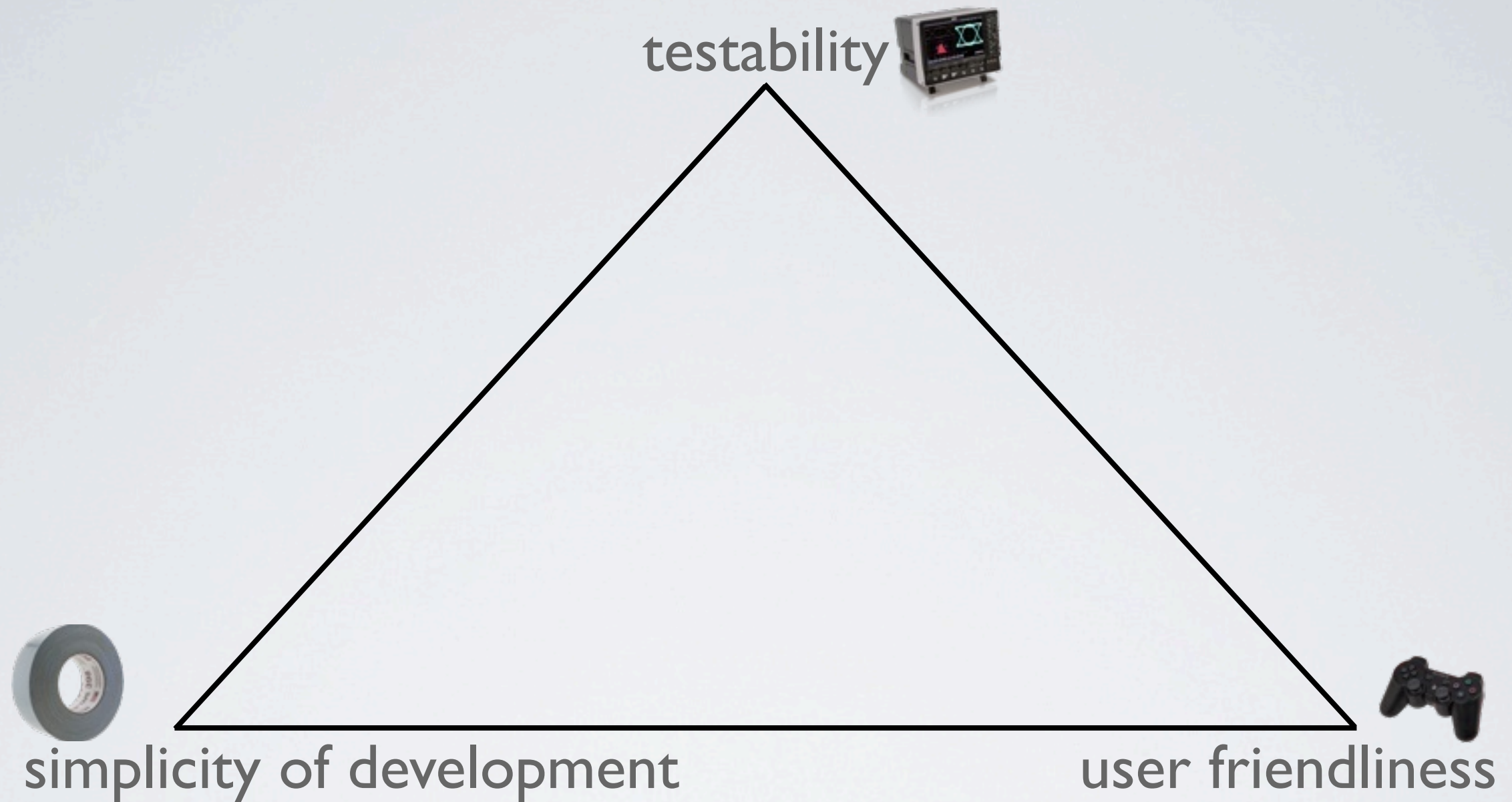
possible issues. exceptions



example application

PROCESS PICKINGLIST

show and tell



THE COMPETITION





Microsoft®

Visual Basic.net

THE COMPETITION



TRADEOFFS

user friendliness



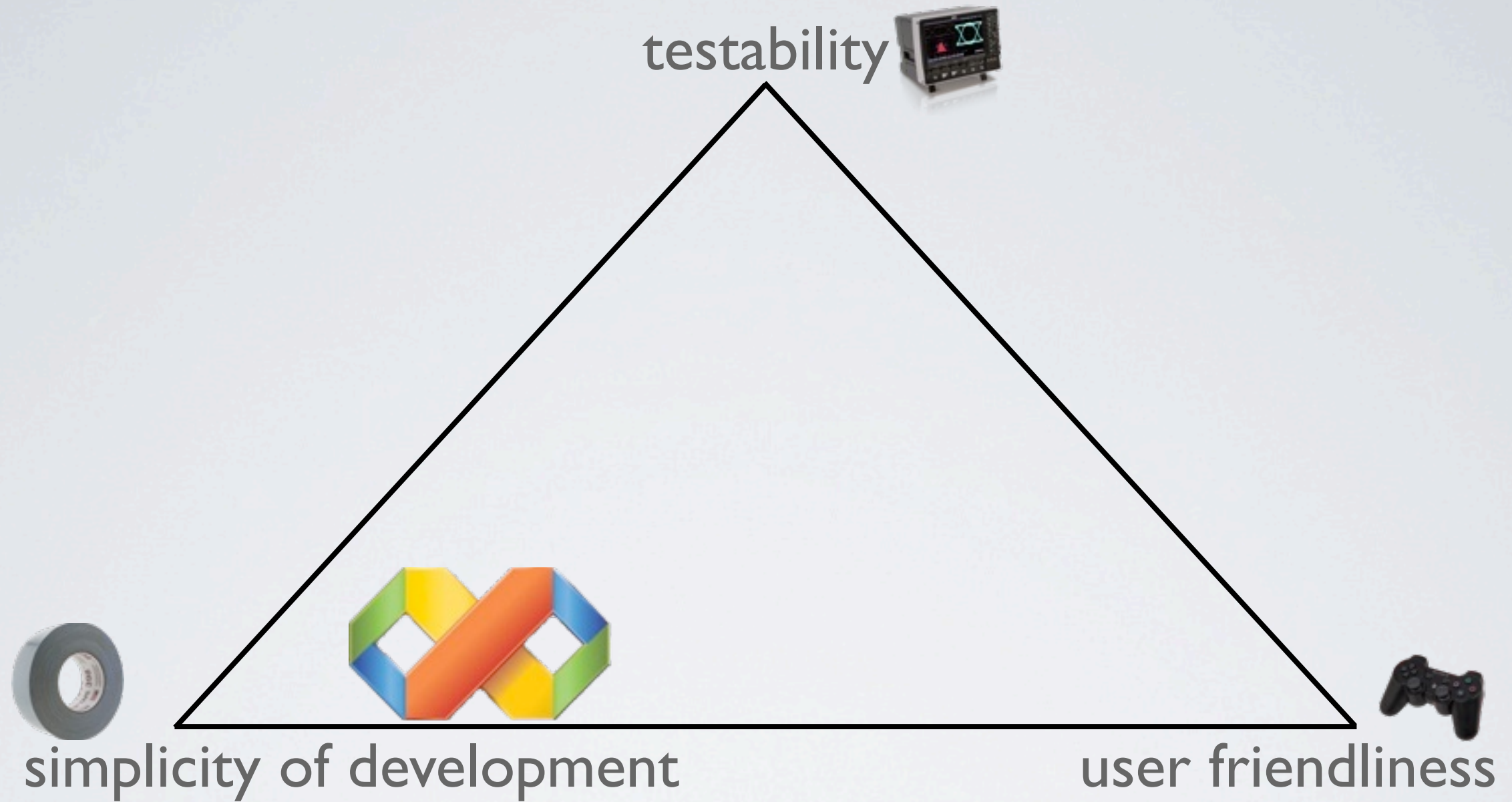
TRADEOFFS

testability



TRADEOFFS

simplicity of development



BALANCE

visual basic



THE COMPETITION



TRADEOFFS

user friendliness



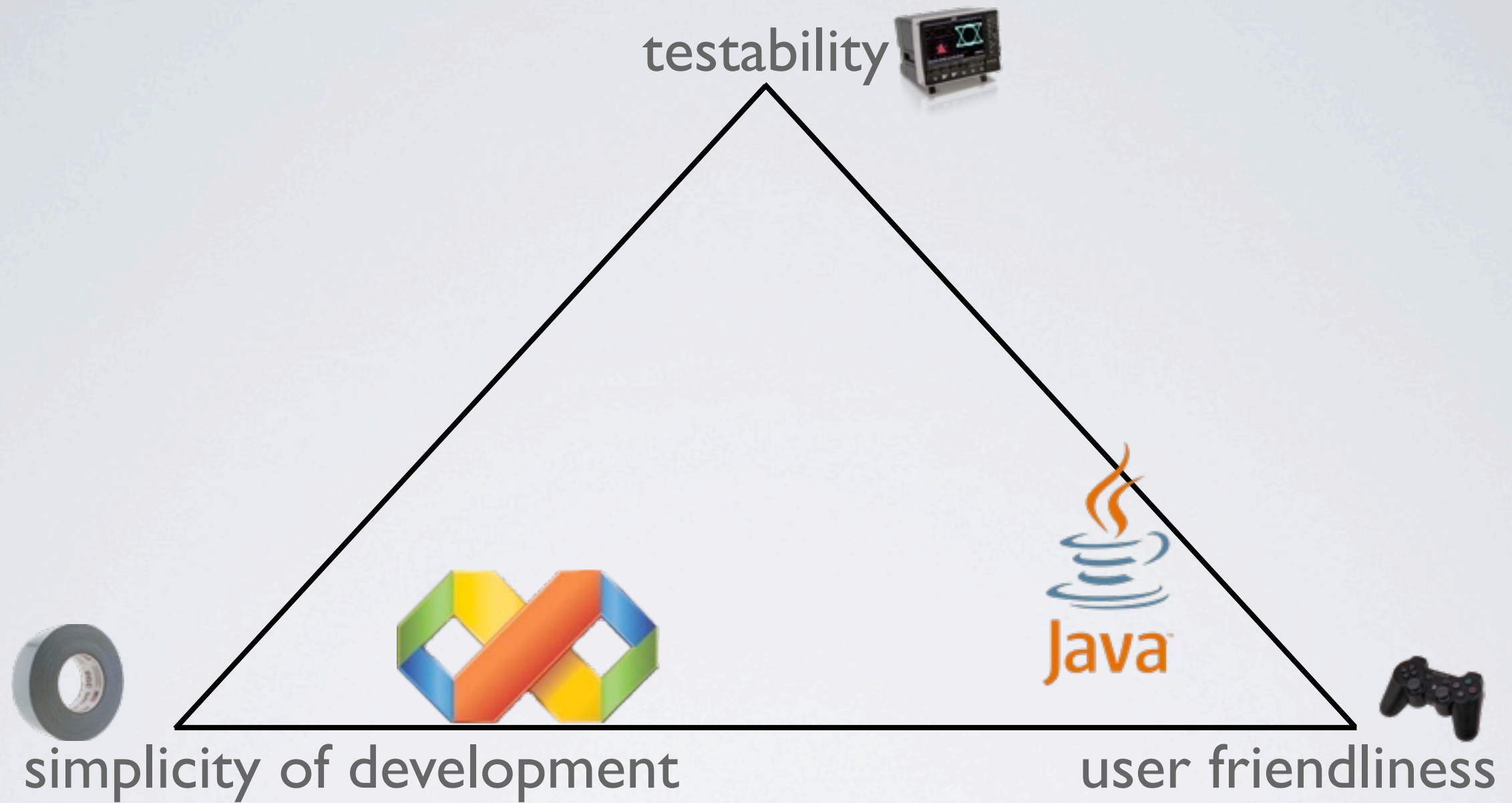
TRADEOFFS

testability



TRADEOFFS

simplicity of development



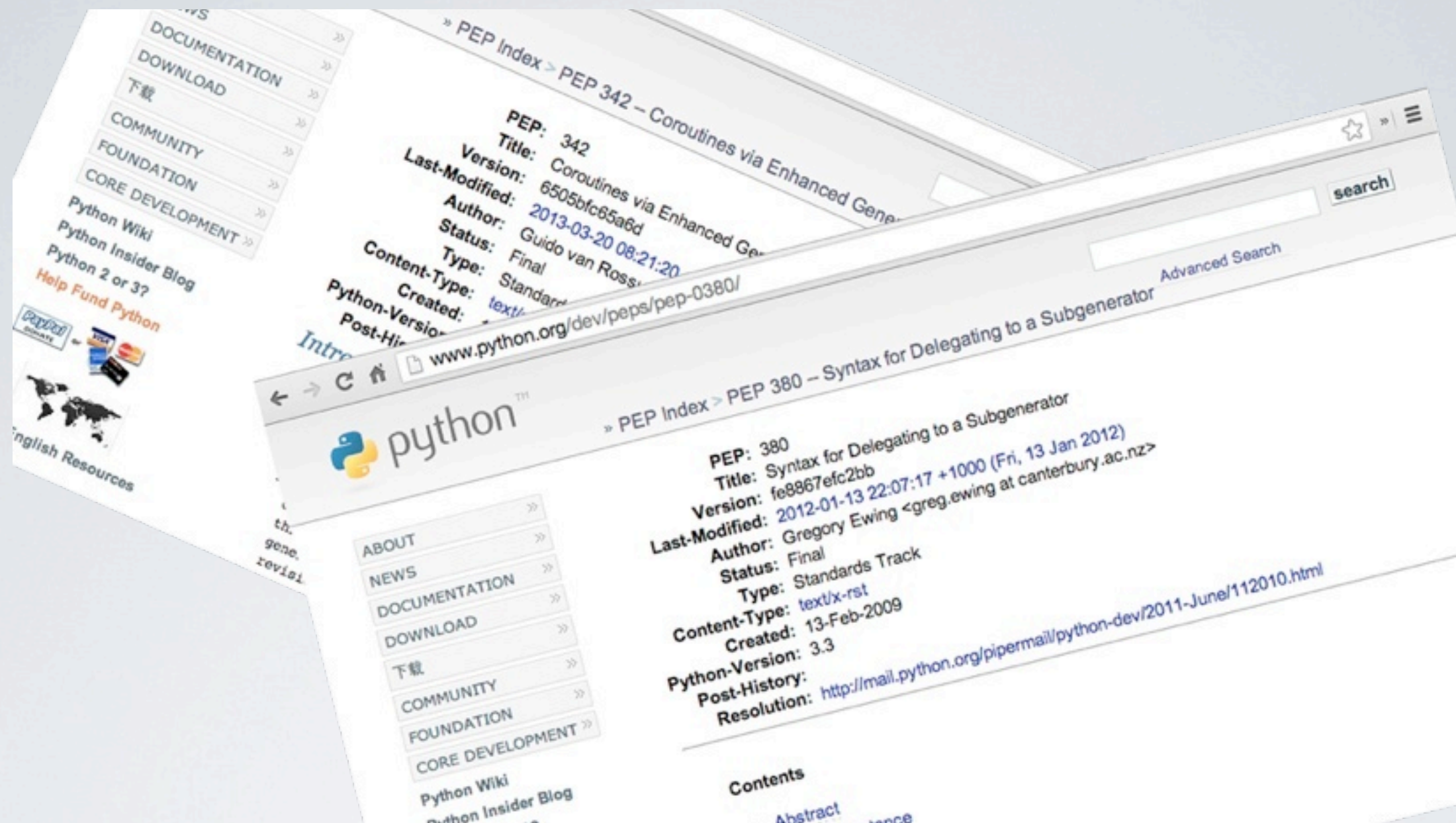
BALANCE

java



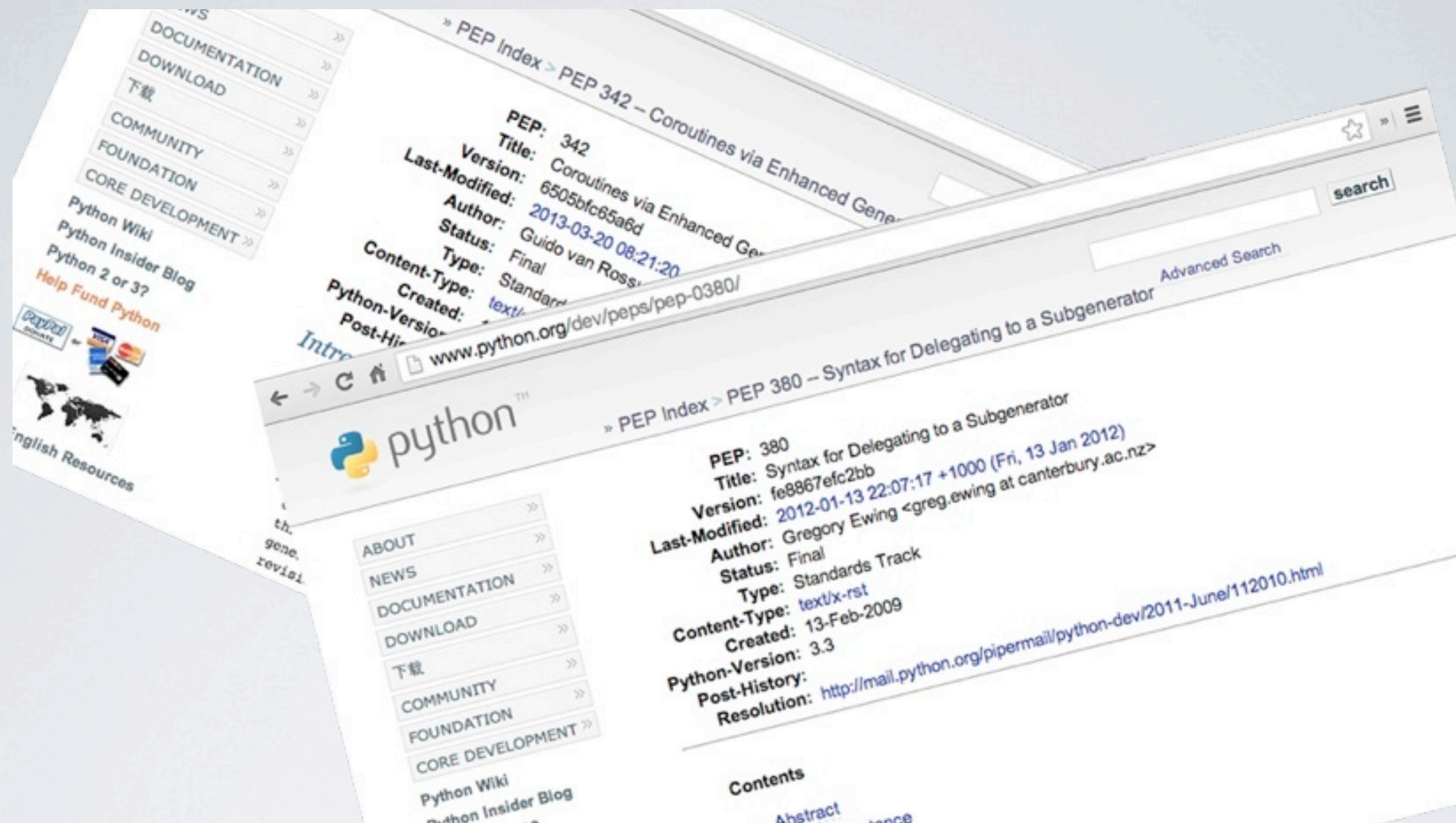
PYTHON

unique language features



PEP 255

Simple generators



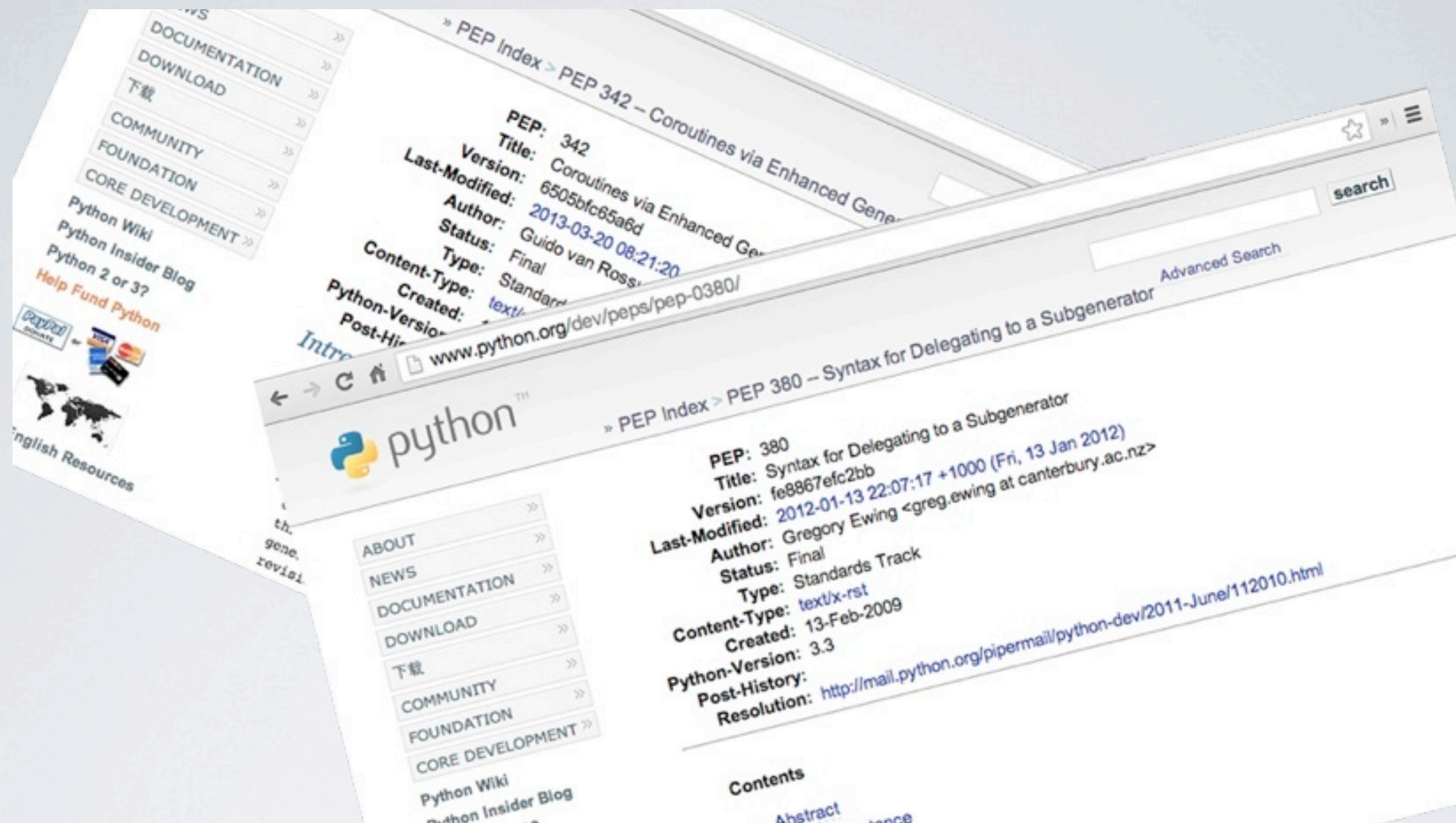
PEP 342

Coroutines via Enhanced Generators

1. Redefine `yield` to be an expression
2. Add a new `send()` method
3. Add a new `throw()` method
4. Allow `"yield"` to be used in `try/finally` blocks

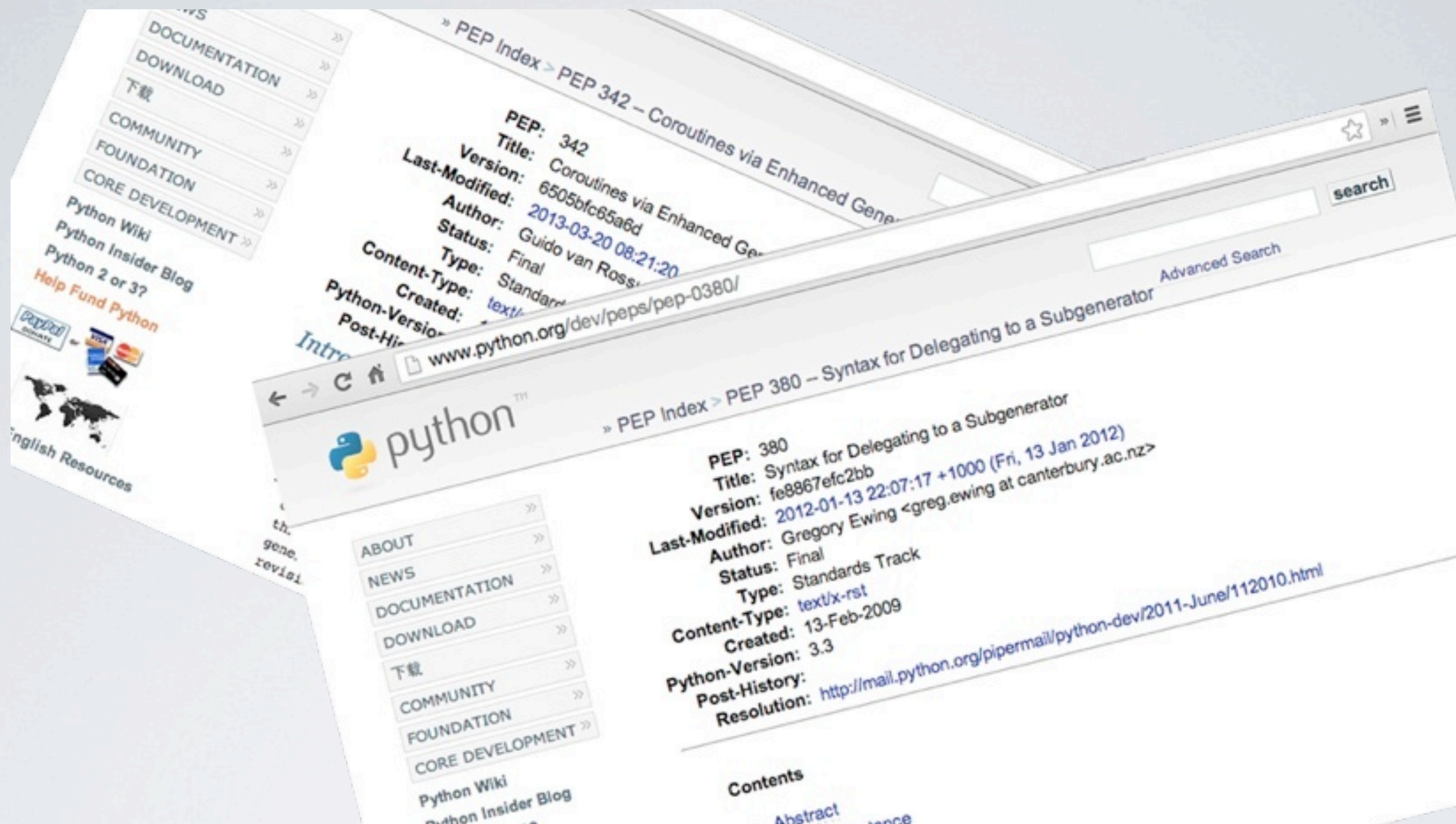
PEP 342

Coroutines via Enhanced Generators



PEP 342

An application as a series of action steps



PEP 342

EXAMPLE APPLICATION

select a file

- ChangeObject
- ChangeObjects
- PrintChart
- PrintPreview
- PrintHtml
- PrintJinjaTemplate
- OpenFile
- OpenStream
- OpenJinjaTemplate
- CloseView
- MessageBox
- Refresh
- OpenFormView
- ShowPixmap
- ShowChart
- SelectFile
- SelectObject

ACTION STEPS

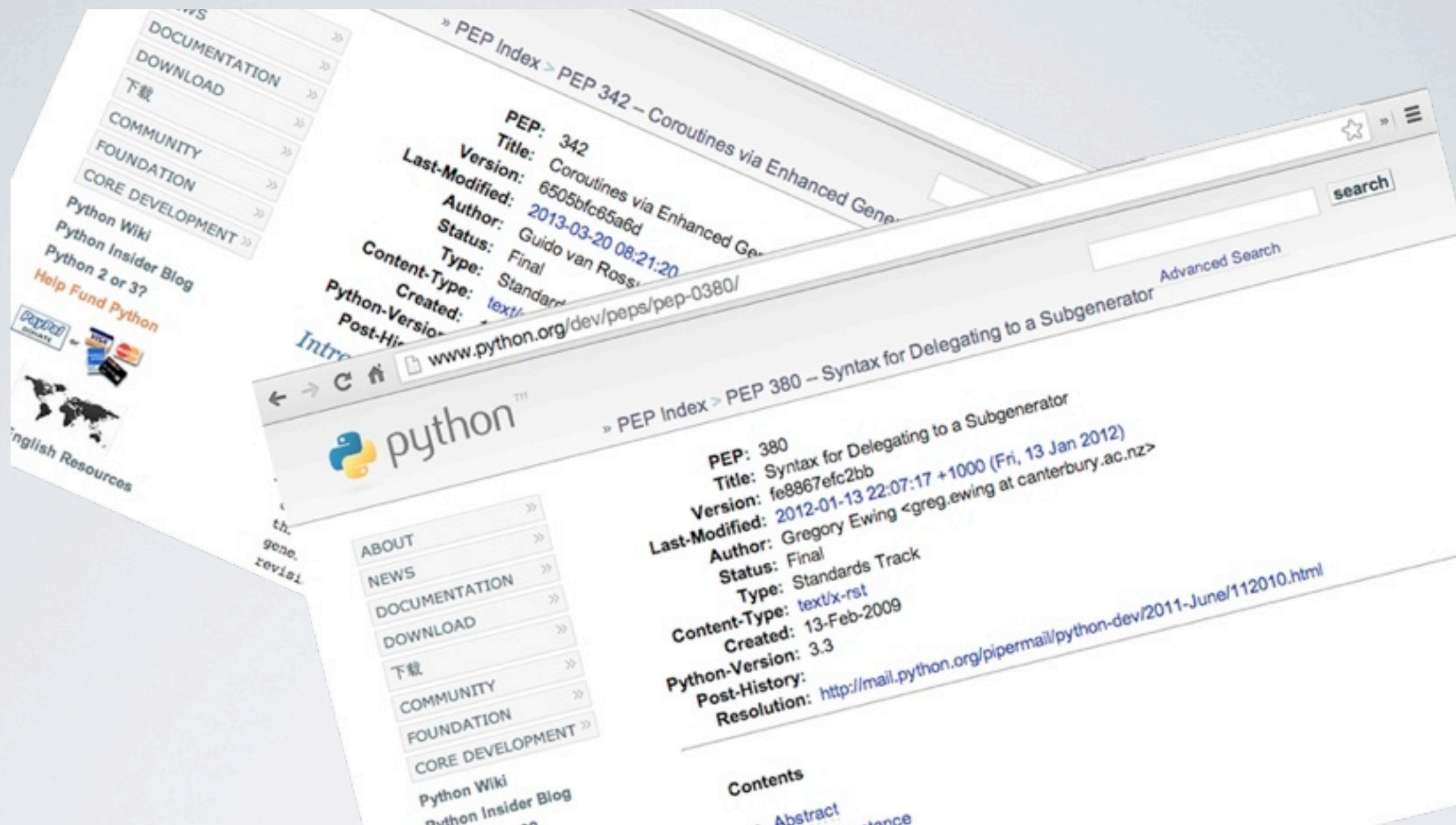
out of the box in Camelot





EVENT LOOP INTEGRATION

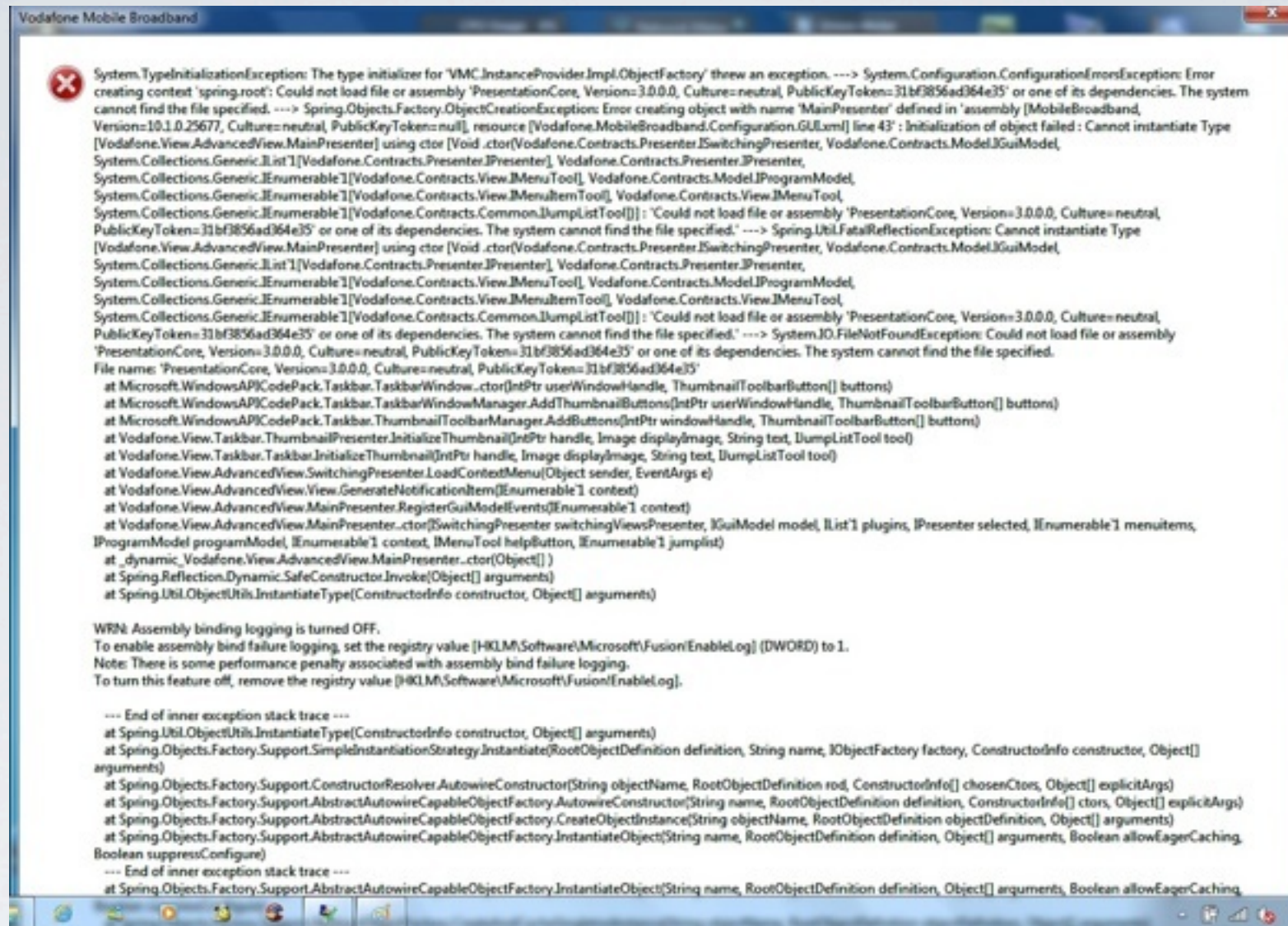
Using multiple threads to prevent freezes



event loop integration

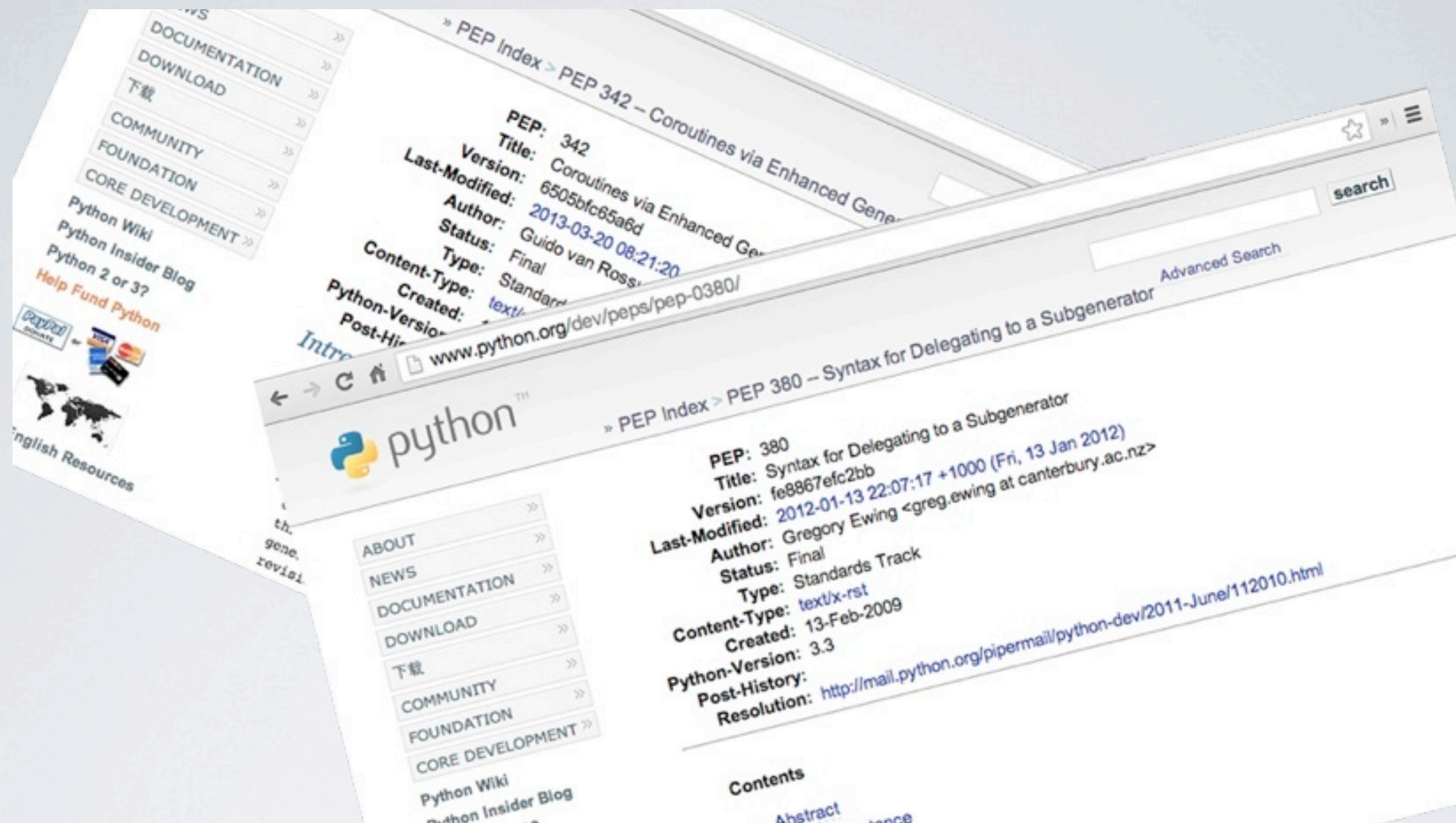
EXAMPLE APPLICATION

import data



EXCEPTIONS

wait ... what?

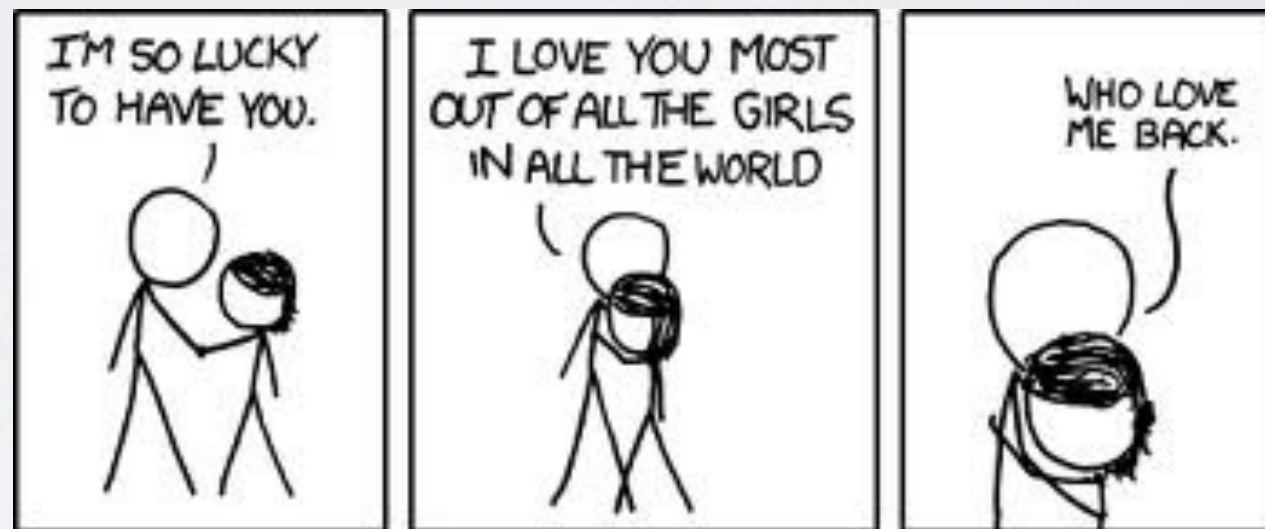


EXAMPLE APPLICATION

user cancels import

- get_collection
- get_selection
- get_object

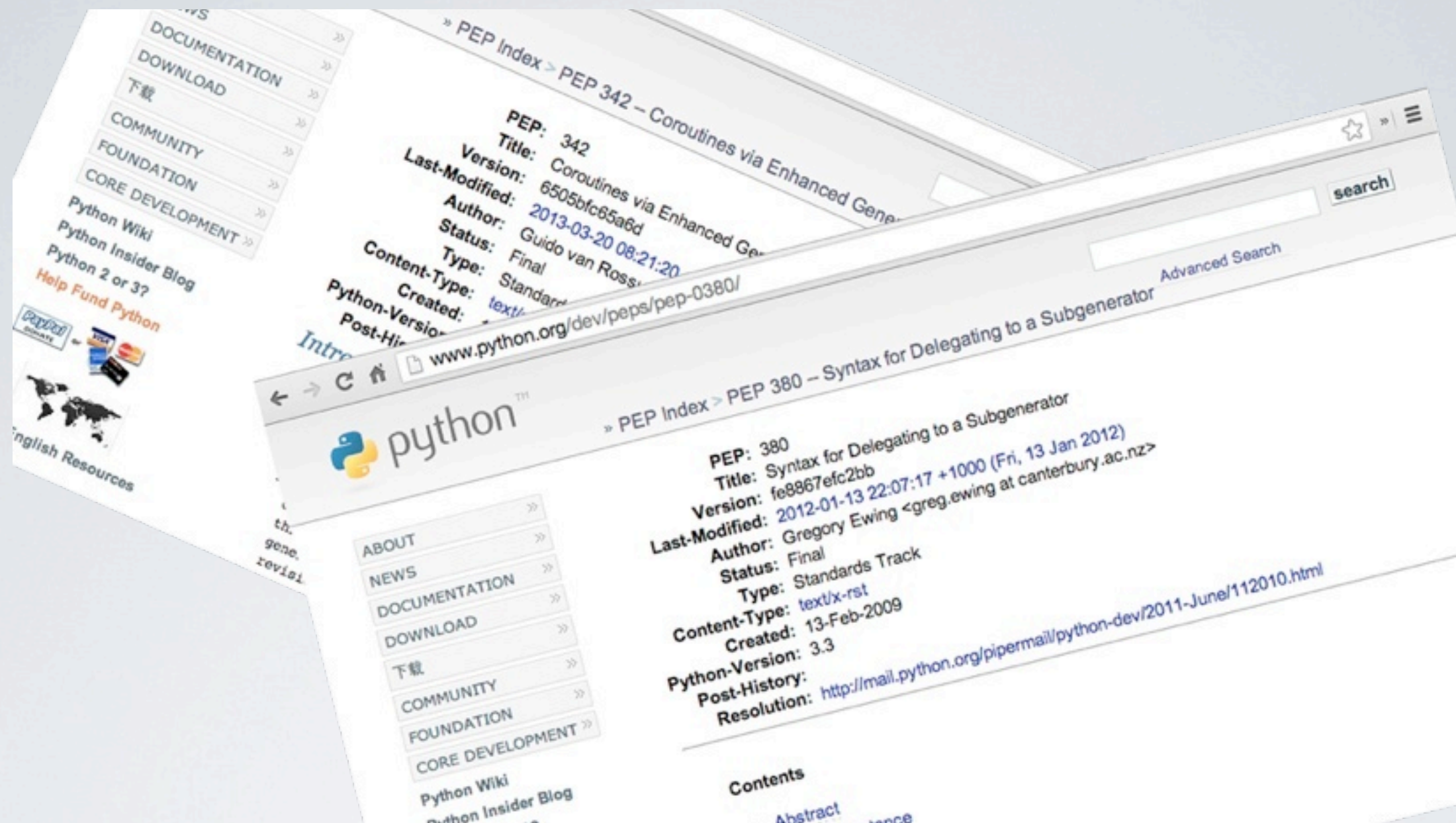
obligatory xkcd comic



WHAT'S THE CONTEXT



TESTABILITY

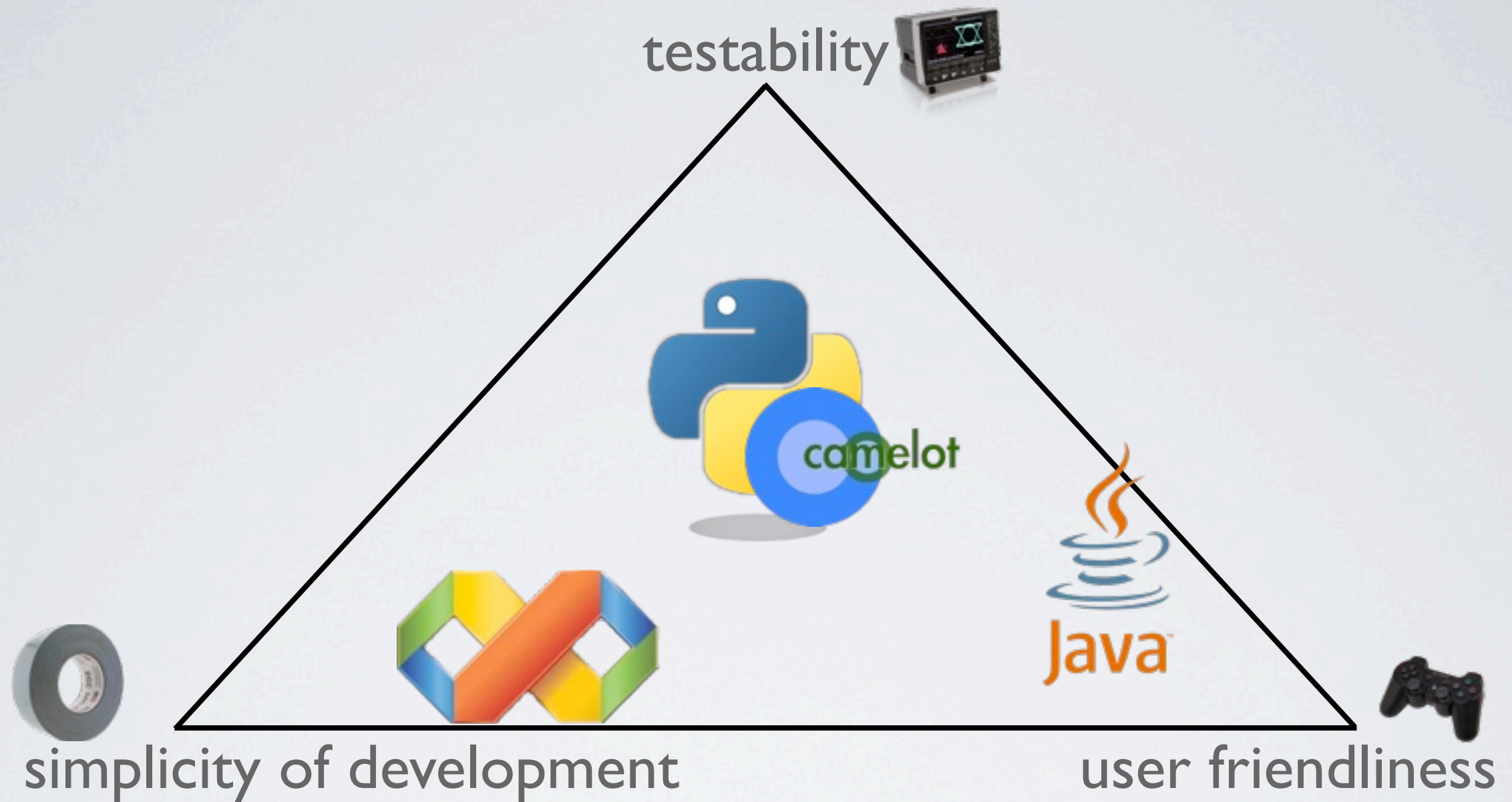


UNITTESTING

example application in action and code ...



THE SWEET SPOT



BALANCE

python (camelot)

The image is a screenshot of the Python.org website, specifically the page announcing Python 3.3. The top navigation bar includes links for Python, PSF, Docs, PyPi, Jobs, and Education. Below this is the Python logo and a search bar. A secondary navigation bar contains links for About, Downloads, Documentation, Community, Success Stories, Blog, and Events. The main content area features a code snippet on the left and a text block on the right. The code snippet shows a generator function `g(x)` that yields from `range(x, 0, -1)` and `range(x)`, followed by a call to `list(g(5))` which returns `[5, 4, 3, 2, 1, 0, 1, 2, 3, 4]`. The text block, titled "Python 3.3", explains that PEP 380 adds the `yield from` expression, allowing a generator to delegate part of its operations to another generator. At the bottom, a footer states: "Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)".

Python

PSF

Docs

PyPi

Jobs

Educational

python™ **BETA**

Search **GO** A A Soc

About Downloads Documentation Community Success Stories Blog Events

```
1 >>> def g(x):
2 ...     yield from range(x, 0, -1)
3 ...     yield from range(x)
4 ...
5 >>> list(g(5))
6 [5, 4, 3, 2, 1, 0, 1, 2, 3, 4]
```

Python 3.3

PEP 380 adds the `yield from` expression, allowing a generator to delegate part of its operations to another generator. This allows a section of code containing `yield` to be factored out and placed in another generator. Additionally, the subgenerator is allowed to return with a value, and the value is made available to the delegating generator.

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

PEP 380

delegating to a subgenerator



THE NEXT ITERATION OF YOUR GUI

PEP 342 (Coroutines via Enhanced Generators)
PEP 380 (Syntax for Delegating to a Subgenerator)



conceptive.be @erikjanss
shore.be @jeroendierckx



*Thanks !
Questions ?*

THE NEXT ITERATION OF YOUR GUI

PEP 342 (Coroutines via Enhanced Generators)
PEP 380 (Syntax for Delegating to a Subgenerator)

We're hiring !



conceptive.be @erikjanss
shore.be @jeroendierckx