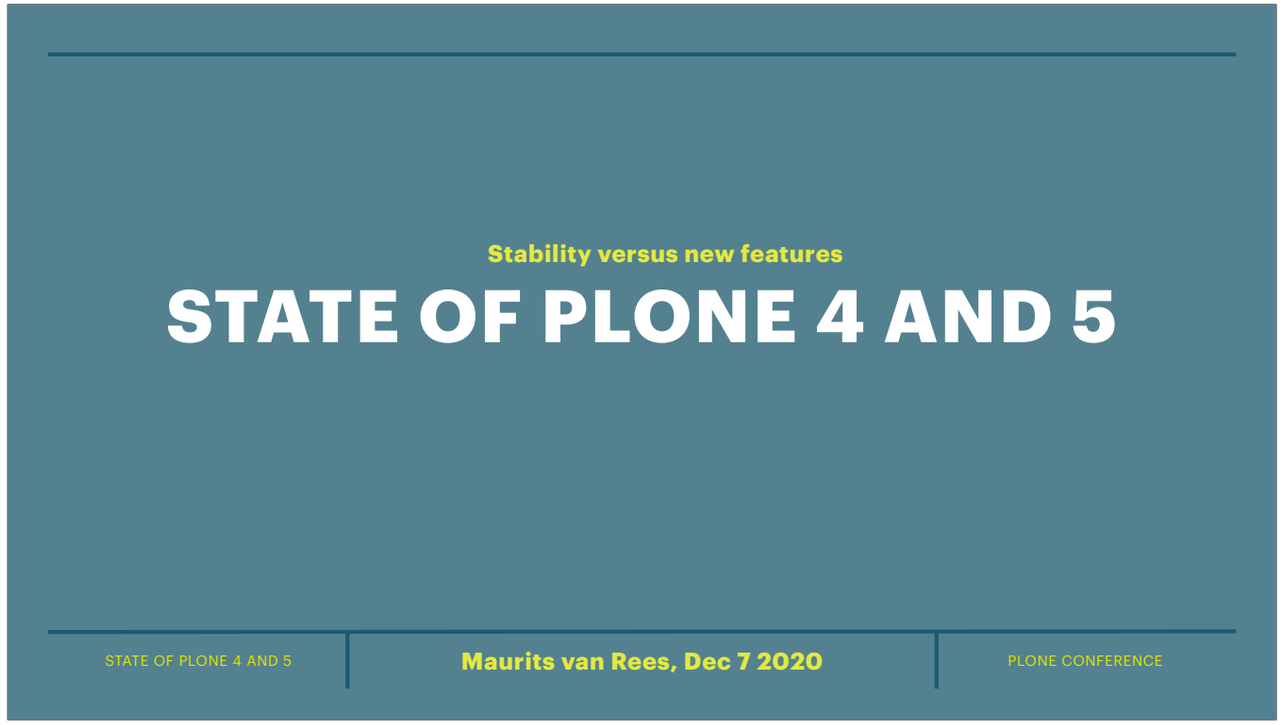
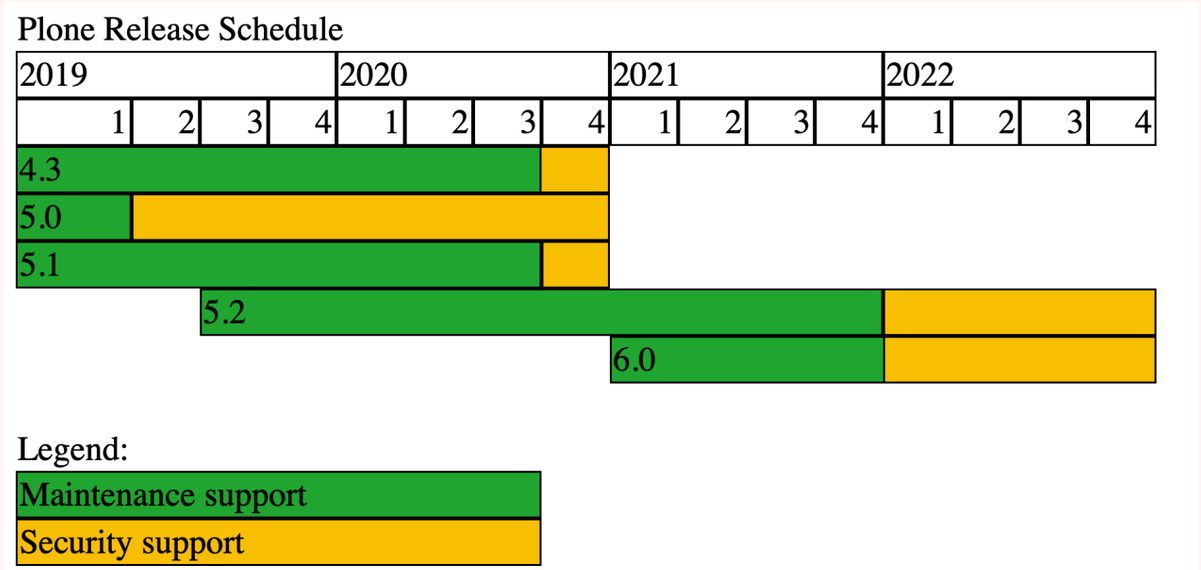


Hello everyone. I will be presenting the State of Plone 4 and 5.
As a general theme throughout this talk I will focus on stability versus new features.

WHO AM I?

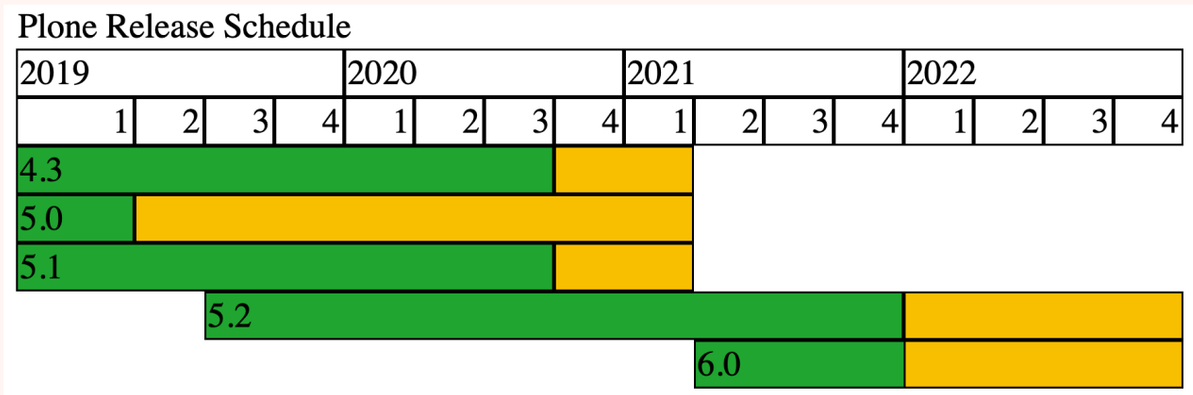
- Maurits van Rees
- Netherlands -> Belgium
- Zest Software, Netherlands
- 15 years Plone
- Security Team
- Release Team

I have lived in The Netherlands for most of my life.
Two years ago I moved to Belgium, where I got married with Fiorella, the lovely lady on the right. And there is my daughter Nika and since 15 October I have a sonTobias.
Since the beginning of 2020 I am the second release manager.
The idea is that I focus on the current versions of Plone, getting regular, stable releases out.
Since then I made four Plone releases: 4.3.20, 5.1.7, 5.2.2 and 5.2.3. I will talk a bit about those.



https://plone.org/download/release-schedule

But one of the first things that I did, was to create a Plone release schedule. This was inspired by the Django release schedule. Django has it a bit easier, because its core is only one package, where Plone has lots. So the Plone release schedule is by nature less predictable.
You can fine the schedule at https://plone.org/download/release-schedule
This is the place to go when you are looking for information on what is the latest version within a Plone release, and is this Plone release still supported, and how long will it be supported.
The 4.3, 5.0 and 5.1series get security support until the end of this year or until Plone 6 comes out. So how will that look?



https://plone.org/download/release-schedule

Let’s say Plone 6.0 is released in a few months.

New features vs stability

We want stable releases. But we also want new features. (Explain picture)
Rule: once we have a Plone 6.0 beta release, no new features should be added.
But most Plone 4 and 5 bugfix releases have had some small new features.
So does this rule only count for major new features, added via the PLIP process?

Does this rule count for dependencies? Do we want only bugfix releases for e.g. zest.releaser, pep517, Pillow? Or is it safe to use all the latest releases, as long as our tests do not break?
And when is something a new feature anyway? I will show some examples.
I will now talk about Plone 4.3, 5.0, 5.1, 5.2, with notes about their stability.

PLONE 4.3

- 4.3.0: April 2013
- 4.3.20: August 2020
- ~~Python 2.6~~
- ~~Maintenance support~~
- Security: until Dec 2020 or Plone 6

4.3 started seven years ago, which really meant it was time to say goodbye.

PLONE 4.3.20

- January 2020 hotfix integrated
- Products.isurlinportal, hotfix-like
- Removed broken X-XSS-Protection header from classic and unstyled themes.

Added Products.isurlinportal, with security hardening of urls. You can install this package on all Plone 4.3 such, just like a regular security hotfix.

PLONE 5.1

- 5.1.0, March 2018
- 5.1.7: October 2020
- ~~Maintenance support~~
- Security: until Dec 2020 or Plone 6
- Last Plone with Zope 2

Warning: the Zope 2 series is not getting fixes anymore.

PLONE 5.1.7

- Same security fixes as 4.3.
- Lots of bug fixes, including Windows, frontend, translations.
- plone.namedfile: Range support

January hotfix, Products.isurlinport.
Range requests are part of the HTTP specification. It means a browser can ask Plone to only send the first megabyte of a large file or video, and later it asks for the next megabyte.
These improvements are also in 5.2.
plone.scale: direction deprecated, use mode:
direction = down -> mode = contain or scale-crop-to-fit
direction = up -> mode = cover or scale-crop-to-fill
direction = thumbnail -> mode = scale

5.1 STABILITY

- **5.1.4: plone.subrequest 1.8.6 has Python 3 fix. Broke subrequests with VirtualHost rewriting: UnicodeDecodeError.**

Example from a customer. We have a control panel where a site administrator could add some links that we show in the footer. We inserted this html snippet using a Diazo rule, which uses plone.subrequest. This worked in English en Dutch, but went wrong in French and Greek.
Solution: create a branch of plone.subrequest that is used for 5.1 and earlier, without the Python 3 fixes.
So in the 5.1 series I only remember one stability problem that was caused by a new feature: Python 3 support.
Other people may remember other problems. And certainly there have been problems in Plone 5 with for example the resource registries or the toolbar or the folder contents. The CMFPlone issue tracker lists about 700 issues. But I think most problems are introduced because someone fixes one bug and accidentally creates another. Sorry, that happens. As release team we cannot prevent this.
But the problems I focus on in this keynote, are because new features are added to a Plone series that should be stable.
TODO: move notes to later slide?

PLONE 5.2

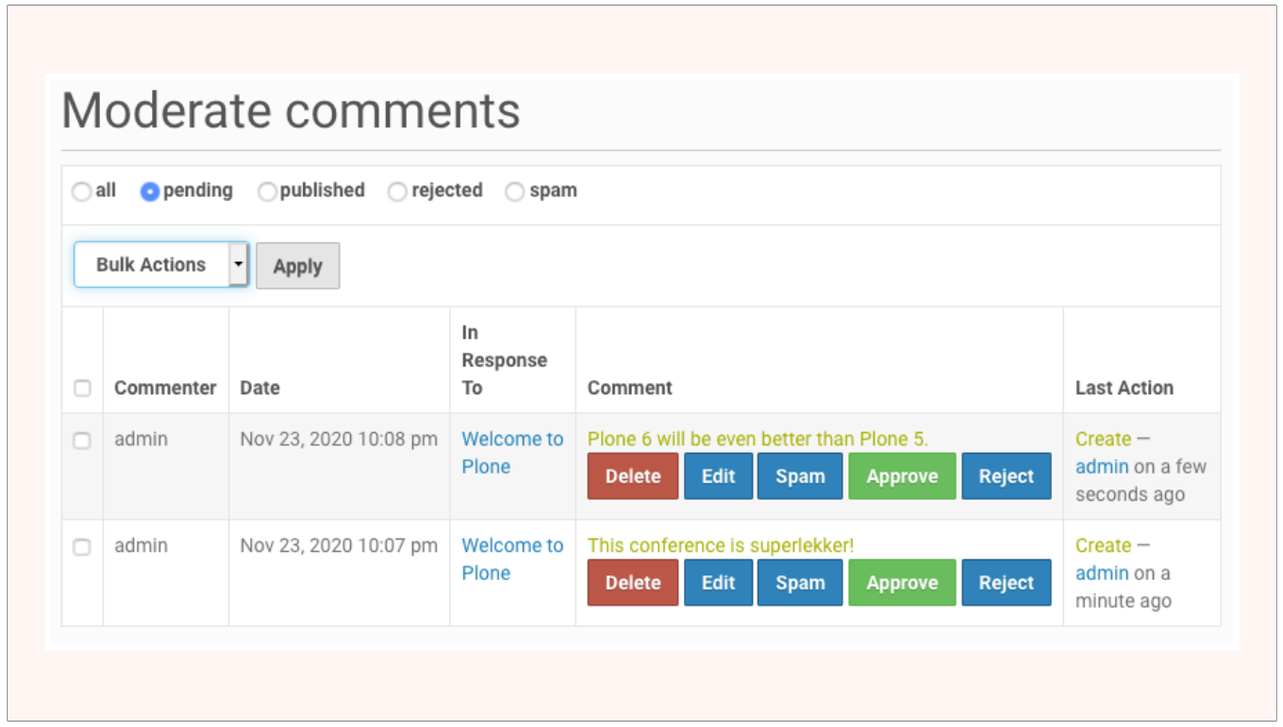
- 5.2.0: July 2019
- 5.2.3: November 2020
- 5.2.4: expected January 2021
- Supports Python 2.7, 3.6, 3.7, 3.8
- Uses Zope 4.
- Upgrade path: 4/5 to 5.2 Python 2 to 5.2 Python 3.
- Maintenance support until December 2021 or until Plone 6.1 comes out, whichever comes last.
- Security support until December 2022 or until Plone 7 comes out, whichever comes last.

This is currently the main version. If you use Plone 5.2 on Python 3, you are in a good position.
Note that Python 2.7 has reached end of life, and will get harder to test and support.

PLONE 5.2.2 + 5.2.3

- Fixes security problems in handling of XML, reported by MisakiKata. (And please upgrade collective.easyform.)
- Zope was upgraded from 4.1 to 4.5. This means the support for WebDav is back.
- zope.interface: more consistent resolution order (5.2.2), fix potential memory leak (5.2.3)
- plone.restapi was upgraded from 6.1 to 6.13, with lots of new features, like endpoints for managing add-ons.
- plone.app.theming: textarea in theming control panel for custom css.
- plone.app.discussion: Extended existing review workflow by states rejected and spam.

- WebDav should work about as good as it used to in Zope 2. For some people this is enough, for others not. I will talk more about Zope in a minute.
- zope.interface was upgraded from 4.6.0 to 5.0.2. This has performance fixes that may have backwards incompatible changes. This makes the inheritance and method resolution order for interfaces more logical and more in line with how it works for 'normal' classes. In corner cases there can be subtle differences when there are several utility or adapter registrations for similar interfaces. For example, when looking for browser view X and there are two such views, Zope looks for the most specific matching interface, and this may have changed.



In the discussion control panel you enable comment moderation.
Then you get more options on the Moderate Comments page.
You can use this to have one reviewer mark a comment as spam or reject it, and have another go through the spam and rejection lists for a second opinion.
Or you could hook this up to automatic spam detection and have a human editor make the final judgement.

5.2 STABILITY

- Zope 4.4 had changes to the the logic around template engines, more fixes in 4.5.
  - BAD: python:test(condition, ‘yes’, ‘no’) GOOD: python: ‘yes’ if condition else ‘no’
  - BAD: python:repeat[‘widget'].index() GOOD: repeat/widget/index
  - repeat/widget/length is broken. (Fixed in Zope 4.5.3, Plone 5.2.3)

- Zope 4.4 had changes to the the logic around template engines. There should now be less differences in behavior between various kinds of templates (Zope/Acquisition-aware or not) or which template engine is used (zope.pagetemplate or chameleon). There were some problems due to this change, but all should have been fixed in Zope 4.5. Or so we thought. The 'test' keyword is gone. Replace python:test(condition, 'yes', 'no') with python: 'yes' if condition else 'no'. In repeat statements, python:repeat['widget'].index() must be replaced with repeat/widget/index path statement. repeat/widget/length is broken.

SEMANTIC VERSIONING?

- plone.subrequest: 1.8.6 improved Python 3 compatibility, broke subrequests in 5.1
- z3c.autoinclude 0.40:
  - breaking change: drop Python 3.4 support
  - new feature: add support for 3.8.
  - no code change
- Zope:
  - 4.1: used in Plone 5.2.1
  - 4.2: WebDav (good)
  - 4.3: WebDav independent of ZServer (good)
  - 4.4: template engine fixes (bad)
  - 4.5: more template fixes (good)
  - 4.5.2: more template fixes , but break Python 2 (good/bad)
  - 4.5.3: fix Python 2 (good)

- plone.subrequest: these are bugfix releases, according to their version numbers.
- z3c.autoinclude 0.40 is used in latest Plone 4.3, 5.1, 5.2.
- z3c.autoinclude: drop 3.4: Plone does not care.
- z3c.autoinclude: add 3.8: Good to have in Plone 5.2, totally safe in 4.3 too because there are no code changes.
- Z3C_AUTOINCLUDE_DEBUG: should be safe, but it is a small risk.
- Zope: Plone 5.2.1 used Zope 4.1. We could have stayed on that version, but then no bug fixes would have come in for the rest of the entire Plone 5.2 series.
- Zope 4.4 added template engine fixes, which sounded innocent enough, but caused breakage.
- 4.5 fixed several of those problems, and 4.5.1 fixed more.
- 4.5.2 fixed even more, but had an unrelated breaking change on Python 2 for http headers. 4.5.3 fixed that.

PLONE 5: FUTURE

- Stability versus features
- No new features at all?
- Zope 4.6+
- TinyMCE 4.7 to 4.9? 5.x?
- jQuery 1.12 to 3?
- Plone 5.3? No.
- Optional newer mockup and plone.staticresources
- Best place for new features: Plone 6

- You could say: 5.2 is a stable branch, so no new features at all. Only bug fixes. If we would have done this from the start, then for 5.2 this would have meant: Zope 4.1.1 so no webdav, less fast zope.interface, no Range support, plone.restapi 6.1 instead of 6.13 so no add-ons endpoint.
- And that is just from the past. Which future changes would we miss then?
- Zope 4 will still get bug fixes and some new features
- Will our TinyMCE still work on modern browsers in half a year?
- Same for jQuery.
- We could think about doing a 5.3 with more adventurous changes. But this distracts from work on Plone 6. And it would be confusing to Plone users to suddenly introduce a new Plone 5 series when Plone 6 is around the corner.
- Keep an eye out for release notes in future 5.2 versions. I note the most dangerous changes there.
- If someone asks: discussion on jQuery, mockup, staticresources: https://github.com/plone/plone.staticresources/pull/102, Framework team okay: https://community.plone.org/t/fwt-meeting-minutes-2020-11-10/13086