Astrophysics software development:

some perspectives, suggestions, and the write-only documentation phenomenon.

Phil Evans

(University of Leicester)

Who am I / why am I giving this talk?

Phil Evans

Key stats:

Name: (See above).

Age: 4.4540 × 10^{-8} Gyr.

Rest mass: 3.6181 × 10^{-29} $M_\odot$ (R_s = 1.0672 × 10^{-25} m).

Average redshift: 0 (despite the t-shirt…)

Publications: Yes.

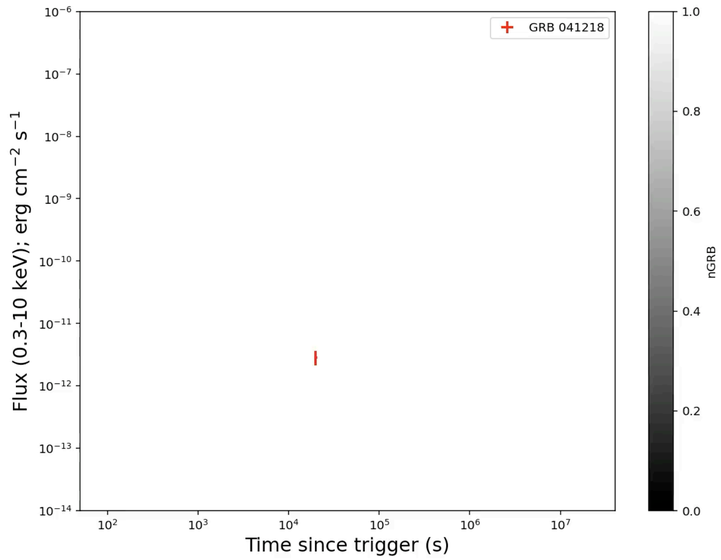
Institute: UNIVERSITY OF LEICESTER

Responsible for the software behind www.swift.ac.uk

Automatic and on-demand analysis of Swift-XRT data.

LSXPS “Living” catalogue and transient system.

Science interests: GRBs, transients, TDAMM.



Phil Evans

Key stats:

Name: (See above).

Age: 4.4540 × 10^{-8} Gyr.

Rest mass: 3.6181 × 10^{-29} $M_\odot$ (R_s = 1.0672 × 10^{-25} m).

Average redshift: 0 (despite the t-shirt…)

Publications: Yes.

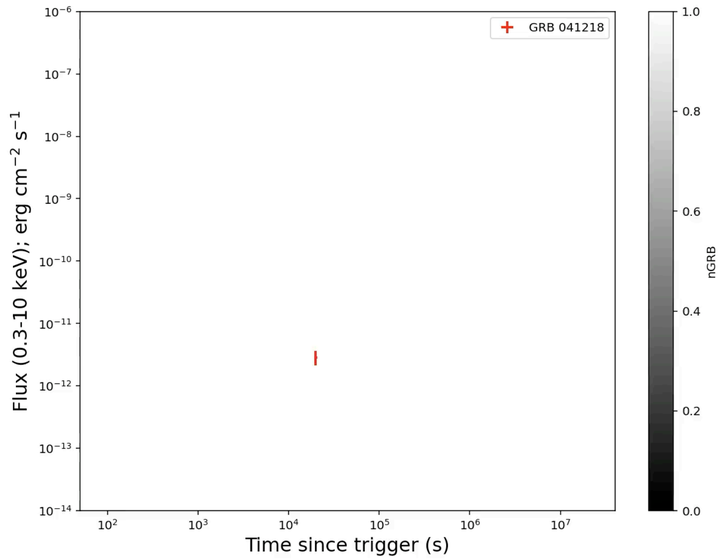
Institute: UNIVERSITY OF LEICESTER

Responsible for the software behind www.swift.ac.uk

Automatic and on-demand analysis of Swift-XRT data.

LSXPS “Living” catalogue and transient system.

Science interests: GRBs, transients, TDAMM.



Who am I / why am I giving this talk?

I'm an X-ray astronomer, who has been writing astrophysical software for 19+ years.

Undergraduate degree in Astrophysics & Computer Science*.

- **Swift:**
  - 2006-2021: Data centre scientist for the UK *Swift* Science Data Centre.
    - From 2014 also been the sysadmin (can I put "full-stack" on my CV?).
  - Since 2021: PI of the UoL Swift project.
  - Since 2023: Lecturer ("Assistant Professor"), allegedly handed over UKSSDC duties.
- **NewAthena**
  - Since 2017: technical lead of the *NewAthena* WFI Instrument Science Centre.
  - (From 2027: lead of the above).

* Computer science "to level 2 only".

My experience / context for my reflections

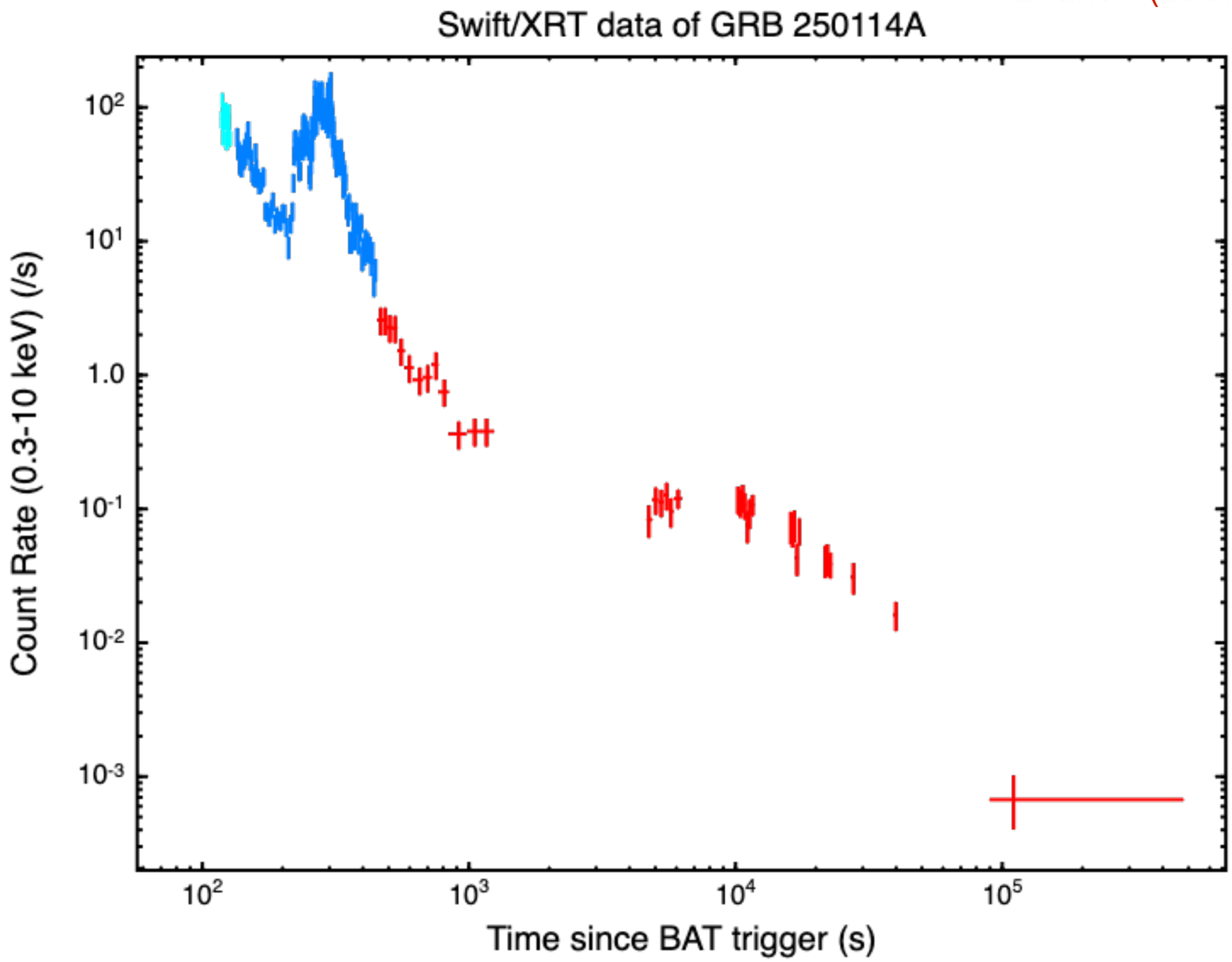
- Building automatic / on-demand analysis tools — *science ready* data products.
  - Optimised extraction, instrumental corrections applied.
  - In (most) cases, more accurate than hand-generation by experts.
- This is an increasingly-common need (e.g. VRO, SKA).

The UKSSDC — a whistle-stop tour

Evans+ (2007), (2009)

What we provide:

- GRBs — automated & public:
  - Lightcurves

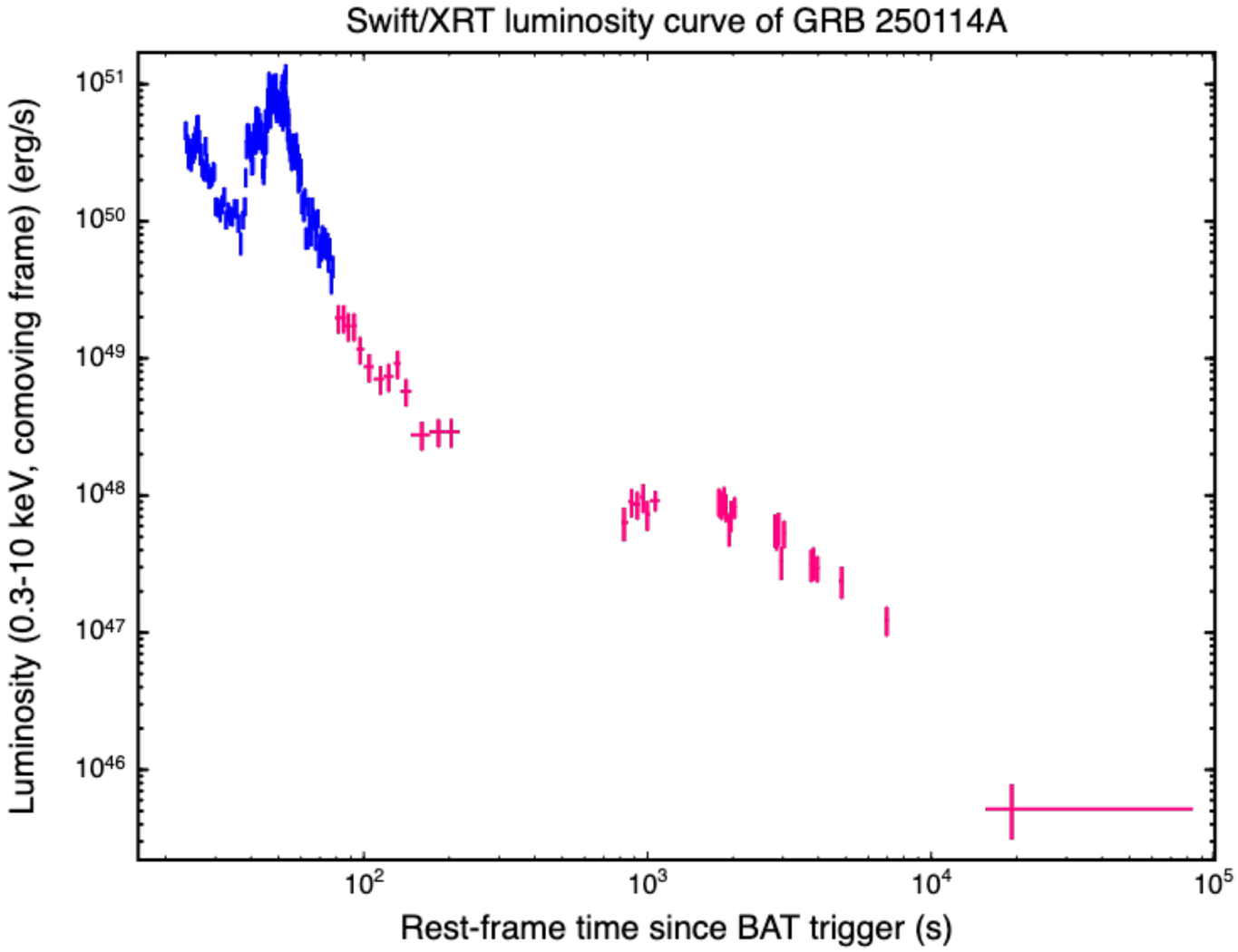


The UKSSDC — a whistle-stop tour

Evans+ (2007), (2009)

What we provide:

- GRBs — automated & public:
  - Lightcurves
    newish with luminosity,
    (when redshift known).

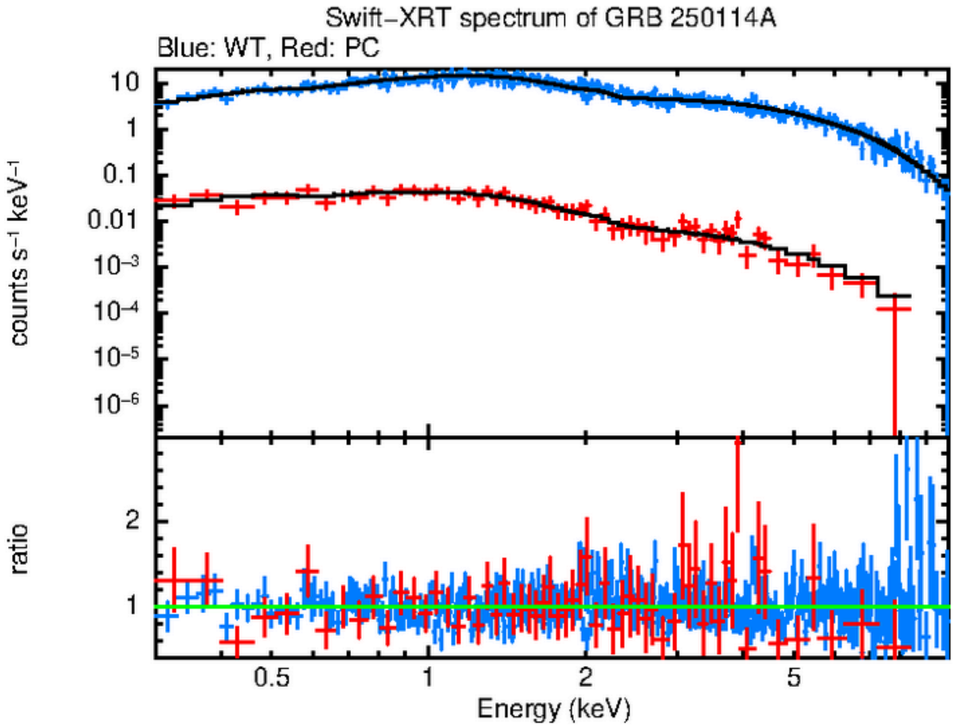


The UKSSDC — a whistle-stop tour

What we provide:

- GRBs — automated & public:
  - Lightcurves,
  - spectra (with fits)

Evans+ (2009)



WT mode. Mean photon arrival: T0+270 s

Insert manually-determined values.

N_H (Galactic)	5.06×10^{20} cm^{-2}
N_H (intrinsic)	$1.8\ (+0.9, -0.9) \times 10^{22}$ cm^{-2}
z of absorber	4.732
Photon index	1.09 (+0.03, -0.03)
Flux (0.3-10 keV) (Observed)	$2.19\ (+0.06, -0.06) \times 10^{-9}$ erg cm^{-2} s^{-1}
Flux (0.3-10 keV) (Unabsorbed)	$2.28\ (+0.05, -0.05) \times 10^{-9}$ erg cm^{-2} s^{-1}

PC mode. Mean photon arrival: T0+8453 s

Insert manually-determined values.

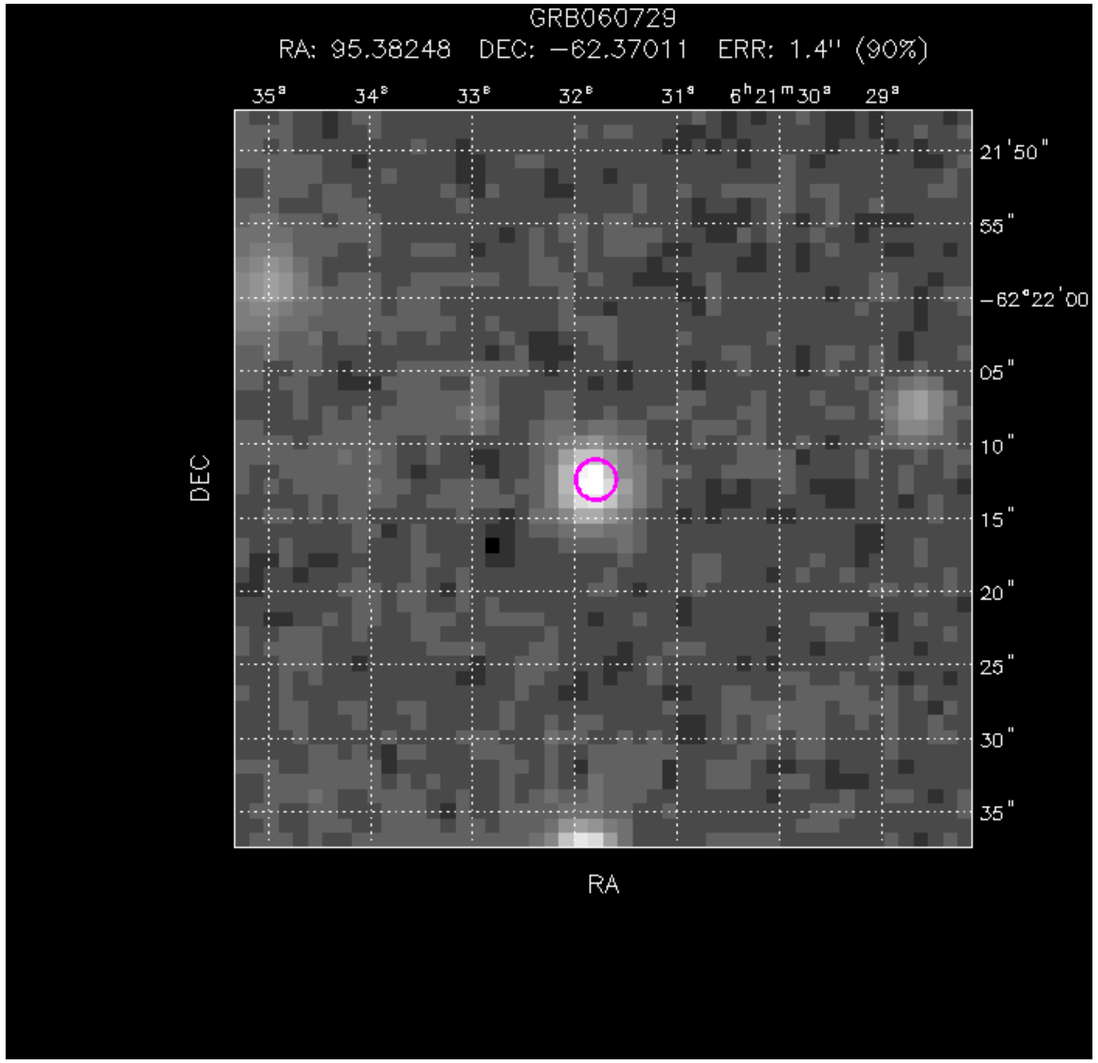
N_H (Galactic)	5.06×10^{20} cm^{-2}
N_H (intrinsic)	$1.0\ (+2.2, -1.0) \times 10^{22}$ cm^{-2}
z of absorber	4.732
Photon index	1.85 (+0.14, -0.13)
Flux (0.3-10 keV) (Observed)	$4.3\ (+0.4, -0.4) \times 10^{-12}$ erg cm^{-2} s^{-1}
Flux (0.3-10 keV) (Unabsorbed)	$4.9\ (+0.4, -0.4) \times 10^{-12}$ erg cm^{-2} s^{-1}

The UKSSDC — a whistle-stop tour

What we provide:

- GRBs — automated & public:
  - Lightcurves,
  - spectra (with fits),
  - (enhanced) positions

Goad+ (2007); Evans+ (2009)



The UKSSDC — a whistle-stop tour

Evans+ (2009)

What we provide:

- GRBs — automated & public:
  - Lightcurves,
  - spectra (with fits),
  - (enhanced) positions,
  - live catalogue

Summary Information

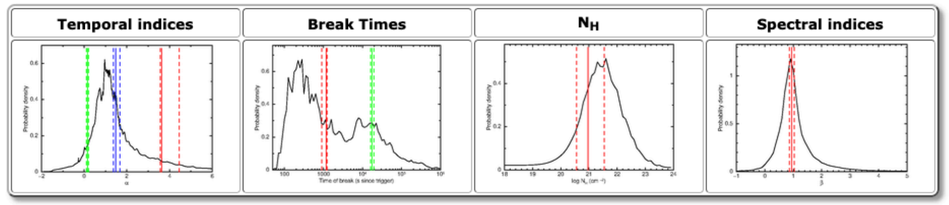
RA:	13h 14m 42.34s	**LC breaks:**	2
Dec:	+05° 01′ 50.8″	**LC type:**	Canonical.
Err:	2.1″	**$N_{H,intr.}$:**	9.25×10^{20} cm
Gal long:	318.045	**Redshift**	2.151
Gal lat:	67.217	**Γ**	1.9

to

α_1	3.63 (+1.35, -0.10)
$T_{b,1}$	1186 (+81, -451) s
α_2	0.16 (+0.09, -0.09)
$T_{b,2}$	1.77 (+0.35, -0.25) × 10^4 s
α_3	1.48 (+0.32, -0.18)

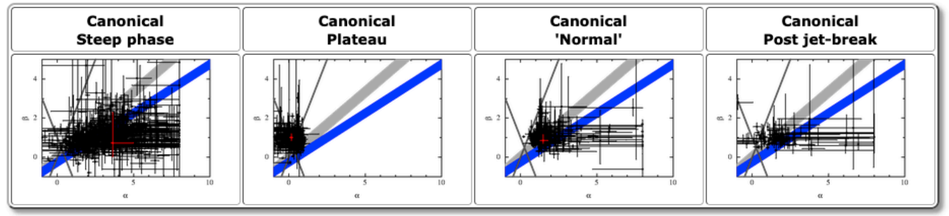
N_{breaks}	χ^2	dof	Comments	Image
0	1138	47		
1	222	45	Break required. (1.13e-14 % probability of chance improvement)	
2	80.4	43	Break required. (3.14e-08 % probability of chance improvement). **This is the best fit.**	
3	54.9	41	Break significance <4-σ (0.04 % probability of chance improvement)	

GRB	RA (J2000)	Dec (J2000)	Pos. Err (″, radius)	#Breaks	Light curve classification	$N_{H,gal}$ ($\times 10^{22}$ cm^{-2})	Redshift
Swift J164449.3+573451	16h 44m 49.95s	+57° 34′ 59.7″	1.4	0	TBD	1.75×10^{-2}	0.354
SGR 1830-0645	18h 30m 41.70s	-06° 45′ 17.2″	1.4	0	No breaks	1.1	
GRB 250129A	13h 14m 42.34s	+05° 01′ 50.8″	2.1	2	Canonical	2.43×10^{-2}	2.151
GRB 250128B	15h 25m 42.10s	-00° 32′ 21.4″	2.7	0	No breaks	8.00×10^{-2}	
GRB 250114B	03h 15m 26.10s	-02° 17′ 38.0″	7.1	0	TBD	7.44×10^{-2}	
GRB 250114A	11h 06m 18.29s	-15° 06′ 41.5″	2.0	2	Canonical	5.06×10^{-2}	4.732
GRB 250108B	13h 25m 18.38s	+25° 36′ 54.8″	2.0	1	One-break	1.16×10^{-2}	
GRB 250108A	13h 51m 41.05s	+26° 12′ 59.4″	3.5	0	No breaks	1.36×10^{-2}	
GRB 250106A	07h 48m 57.30s	+63° 48′ 44.7″	3.0	0	No breaks	4.48×10^{-2}	
GRB 250103B	03h 38m 39.43s	-33° 44′ 55.0″	3.7	0	No breaks	9.99×10^{-3}	1.416



Closure Relationships

The plots below show (α, β) scatter plots for each phase of all Canonical light curves. GRB 250129A is shown in colour; where multiple points appear in a given phase (e.g. WT and PC mode spectra for that phase), the different colours refer to different spectra. α and β are the temporal and spectral energy power-law indices respectively. The shaded regions mark those allowed by the standard closure relationships. These figures correspond to Fig. 10 of Evans et al. (2009). See the closure relationship documentation for more information.

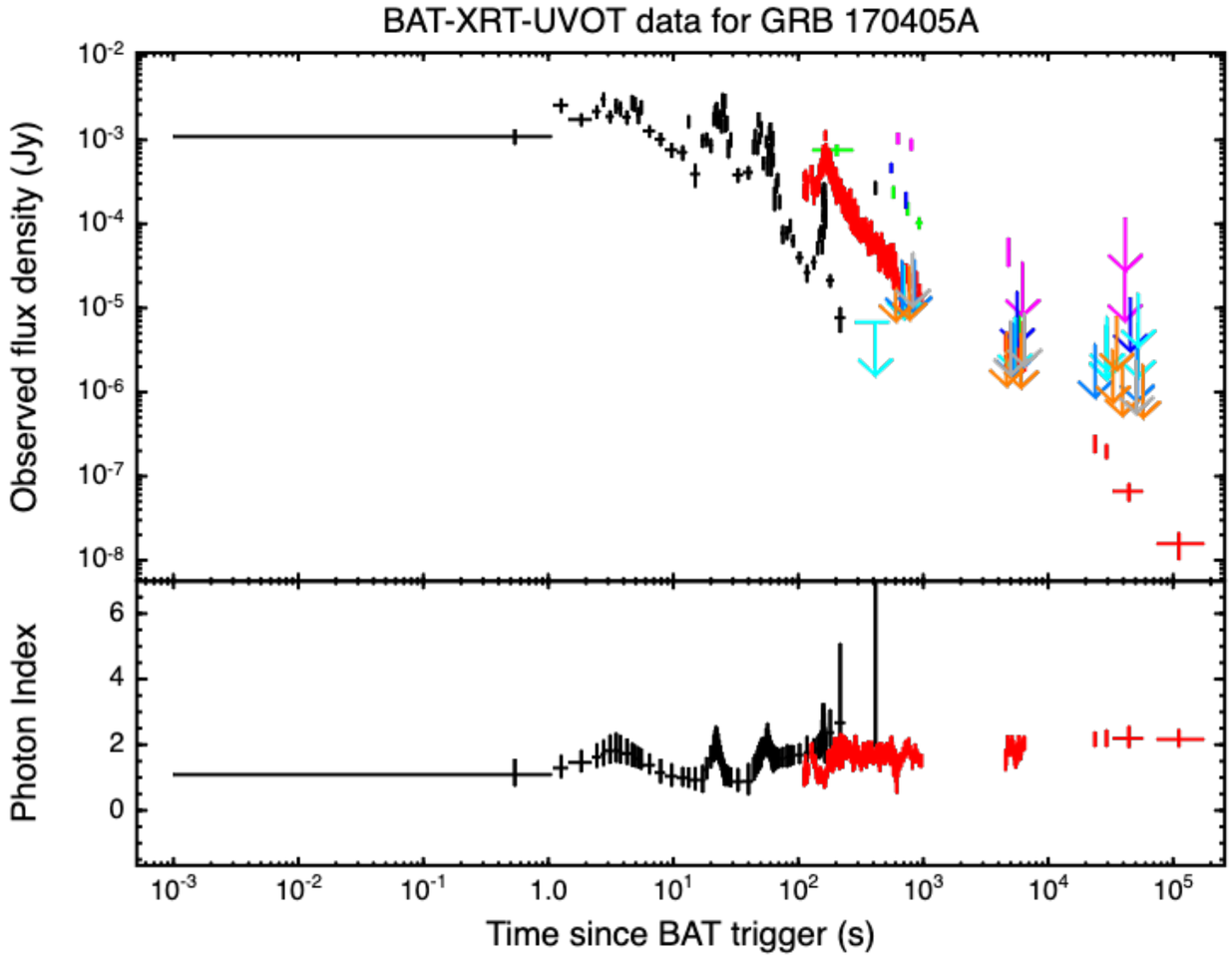


The UKSSDC — a whistle-stop tour

Evans+ (2010)

What we provide:

- GRBs — automated & public:
  - Lightcurves,
  - spectra (with fits),
  - (enhanced) positions,
  - live catalogue,
  - images
  - burst analyser (multi-instrument)



The UKSSDC — a whistle-stop tour

What we provide:

- GRBs — automated & public.
- Source catalogues:

The UKSSDC — a whistle-stop tour

What we provide:

- GRBs — automated & public.
- Source catalogues:
  - 1SXPS (2014)
    https://www.swift.ac.uk/1SXPS
  - 2SXPS (2020)
    https://www.swift.ac.uk/2SXPS
  - LSXPS (2023)
    https://www.swift.ac.uk/LSXPS

Data included	2005-01-01 — 2025-10-06 (408 Ms)
Sky coverage	5,728 deg^2
Detections	2,760,293
Unique sources	346,566
Variable sources[2]	146,390
Uncatalogued sources[2]	131,538

Evans+ (2023a)

The UKSSDC — a whistlestop tour

What we provide:

- GRBs — automated & public.
- Source catalogues:
  - 1SXPS (2014)
    https://www.swift.ac.uk/1SXPS
  - 2SXPS (2020)
    https://www.swift.ac.uk/2SXPS
  - LSXPS (2023) - **with real-time transient detector!**
    https://www.swift.ac.uk/LSXPS

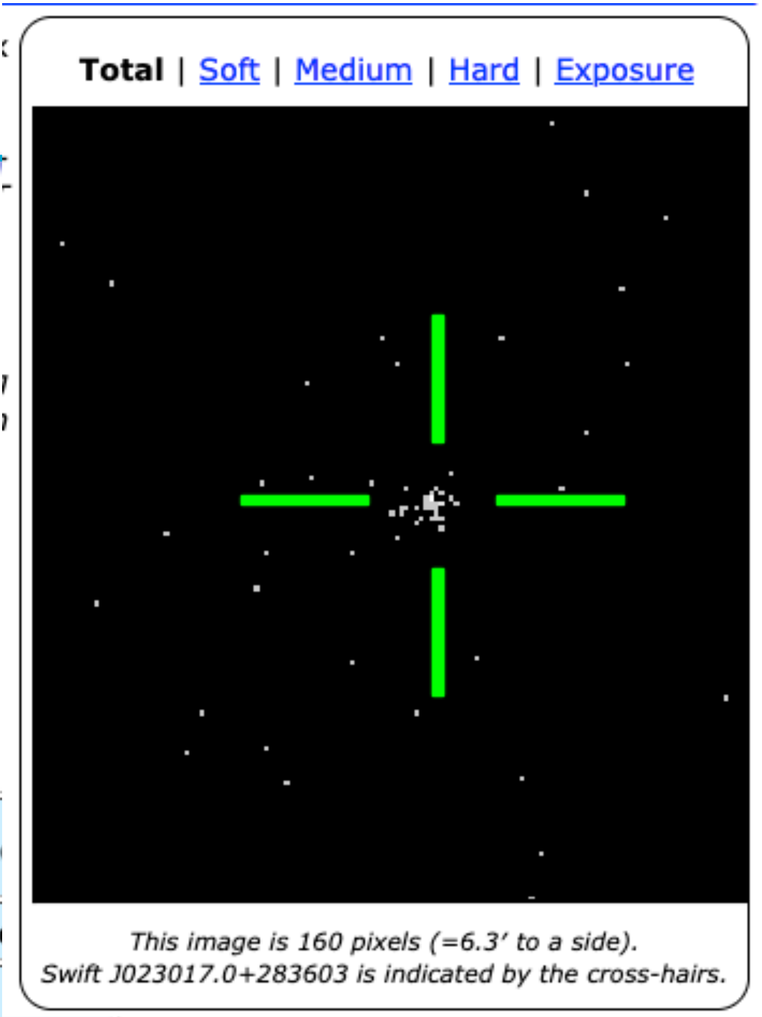
Data included	2005-01-01 — 2025-01-25 (395 Ms)
Sky coverage	5,623 square degrees
Detections	2,537,606
Unique sources	330,170
Variable sources[2]	131,504
Uncatalogued sources[2]	118,238

Evans+ (2023b)

The UKSSDC — a whistlestop

What we provide:

- GRBs — automated & public.
- Source catalogues:
  - 1SXPS (2014)
    https://www.swift.ac.uk/1SXPS
  - 2SXPS (2020)
    https://www.swift.ac.uk/2SXPS
  - LSXPS (2023) - **with real-time transient detector!**
    https://www.swift.ac.uk/LSXPS



Data include	
Sky coverag	
Detections	
Unique sources	330,170
Variable sourcess[2]	131,504
Uncatalogued sources[2]	118,238

Evans+ (2023b)

The UKSSDC — a whistle-stop tour

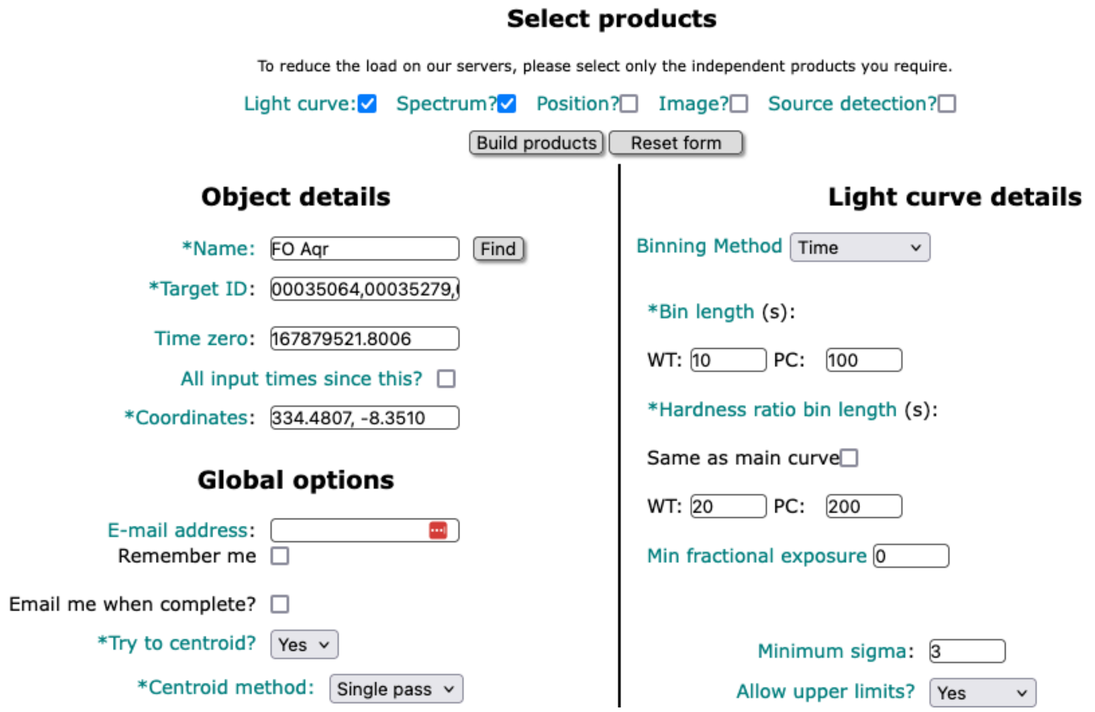
What we provide:

- GRBs — automated & public.
- Source catalogues.
- Analysis of anything

The UKSSDC — a whistle-stop tour

What we provide:

- GRBs — automated & public.
- Source catalogues.
- Analysis of anything*
  https://www.swift.ac.uk/user_objects/
  https://www.swift.ac.uk/API/
  swifttools.ukssdc (via pypi)
- or nothing — source detection.



* Point-sources, seen by XRT; some extreme cases may give imperfect results (although work surprisingly** often).
** Well, I'm always surprised when it works.

Evans+ (2009)

The UKSSDC — a whistle-stop tour

What we provide:

- GRBs — automated & public.
- Source catalogues.
- Analysis of anything*
  https://www.swift.ac.uk/user_objects/
  https://www.swift.ac.uk/API/
  swifttools.ukssdc (via pypi)
- or nothing — source detection.

swifttools.ukssdc.xrt_prods demo

[1]:
```
%load_ext autoreload
%autoreload 2
```

[2]:
```
#from testtools.xrt_prods import XRTProductRequest
from swifttools.xrt_prods import XRTProductRequest
import pandas as pd
me = 'pae9@leicester.ac.uk'
```

Submitting a new job

[3]:
```
myReq = XRTProductRequest(me, silent=False)
myReq.setGlobalPars(getTargs=True,
                    centroid=True,
                    getCoords=True,
                    getT0=True,
                    posErr=0.1,
                    name='GK Per',
                    useSXPS=False
                    )
#myReq.addImage(energies='0.3-10,0.3-1.5,1.51-10', whichData='all')
myReq.addLightCurve(binMeth='obsid',srcrad=57)
myReq.isValid()
```

```
Successfully created a light curve
OK, setting binMeth = obsid
OK, setting srcrad = 57
```

[3]:
```
(True, '')
```

[4]:
```
myReq.submit()
```

```
Job submitted OK, with ID: 196342
Also setting lc useObs = None, because whichData = all
```

[4]:
```
True
```

* Point-sources, seen by XRT; some extreme cases may give imperfect results (although work surprisingly** often).
** Well, I'm always surprised when it works.

Evans+ (2009)

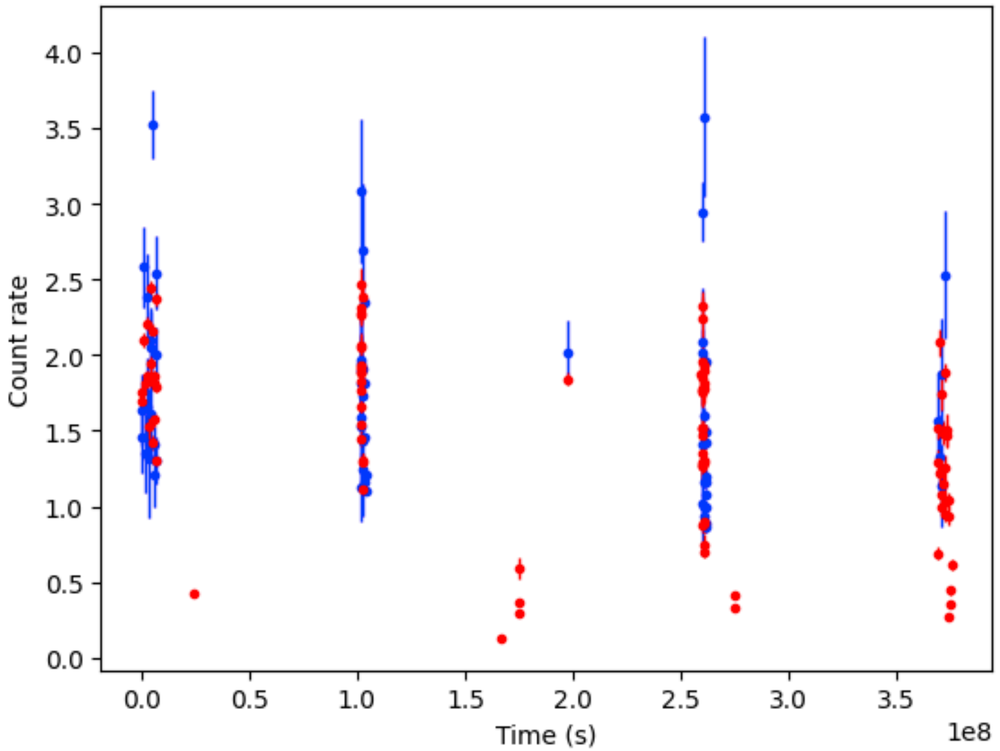
The UKSSDC — a whistle-stop tour

What we provide:

- GRBs — automated & public.
- Source catalogues.
- Analysis of anything*
  https://www.swift.ac.uk/user_objects/
  https://www.swift.ac.uk/API/
  swifttools.ukssdc (via pypi)
- or nothing — source detection.

swifttools.ukssdc.xrt_prods demo

```
[49]: 1 myLC = r.retrieveLightCurve(returnData=True, incbad='yes', nosys='no')
      2 swifttools.ukssdc.plotLightCurve(myLC, whichCurves=("WT_incbad", "PC_incbad"), xlabel="Time (s)
      3
```



* Point-sources, seen by XRT; some extreme cases may give imperfect results (although work surprisingly** often).
** Well, I'm always surprised when it works.

Evans+ (2009)

The UKSSDC — a whistle-stop tour

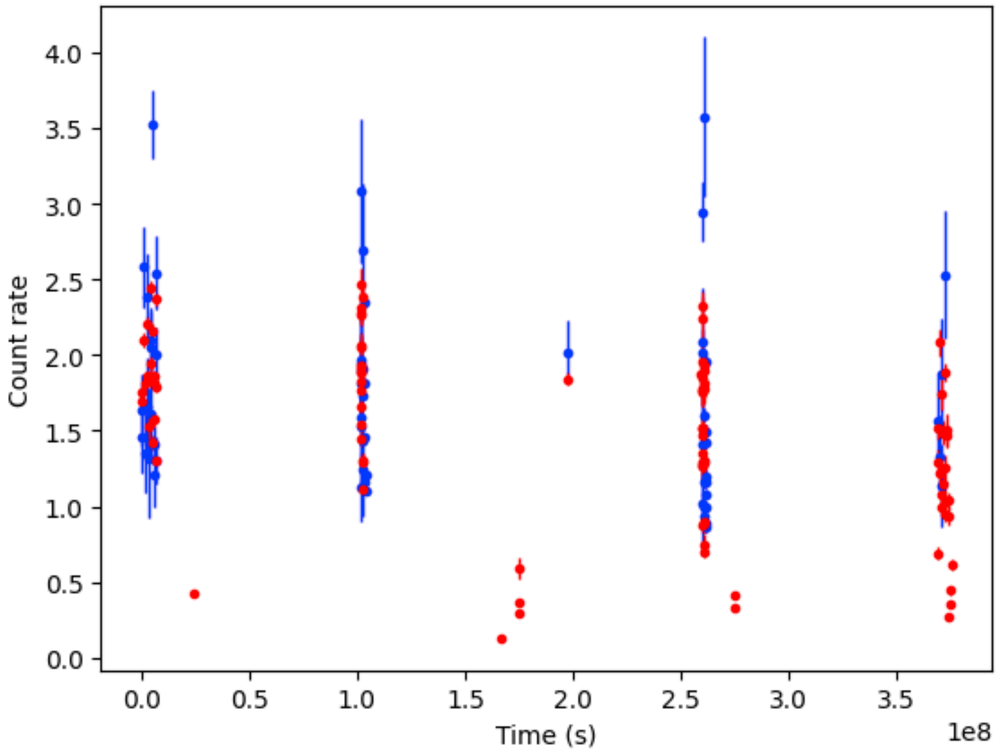
What we provide:

- GRBs — automated & public.
- Source catalogues.
- Analysis of anything*
  https://www.swift.ac.uk/user_objects/
  https://www.swift.ac.uk/API/
  swifttools.ukssdc (via pypi)
- or nothing — source detection.

Produces:
~1400 products/month (web)
~2800 products/month (API)

swifttools.ukssdc.xrt_prods demo

```
[49]: 1 myLC = r.retrieveLightCurve(returnData=True, incbad='yes', nosys='no')
      2 swifttools.ukssdc.plotLightCurve(myLC, whichCurves=("WT_incbad", "PC_incbad"), xlabel="Time (s)
      3
```



* Point-sources, seen by XRT; some extreme cases may give imperfect results (although work surprisingly** often).
** Well, I'm always surprised when it works.

Evans+ (2009)

First warning: tech changes!

- Lead the UK *Swift* Science Data Centre.
  - 2006-2021: Data centre scientist for the UK Swift Science Data Centre.
- Technical lead of the *NewAthena* WFI Instrument Science Centre.

First warning: tech changes!

- Lead the UK *Swift* Science Data Centre.
  - 2006-2021: Data centre scientist for the UK Swift Science Data Centre.
    - “Broadband” internet was a luxury (justified by on-call support).
    - Typical server: dual-core CPU, 4GB RAM. External RAID box for the archive.
    - “Flash” and Java applets were mainstream.
    - People thought IDL was a good idea.
      - (Though some people were trying out ‘Python’)
- Technical lead of the *NewAthena* WFI Instrument Science Centre.

First warning: tech changes!

- Lead the UK *Swift* Science Data Centre.
  - Joined in 2006 as data centre scientist.
    - “Broadband” internet was a luxury (justified by on-call support).
    - Typical server: dual-core CPU, 4GB RAM. External RAID box for the archive.
    - “Flash” and Java applets were mainstream.
    - People thought IDL was a good idea.
      - (Though some people were trying out ‘Python’)
    - 2007: First iPhone goes on sale; xen incorporated into Linux kernel.
    - 2008: first Android phone goes on sale; Python 3.0 released.
    - 2012: 11 UK Cities get 4G mobile internet; automated storage tiering becomes mainstream.
    - 2013: Docker released. Astropy first release. Java applet support wanes.
    - 2014: Kubernetes released.
    - 2019: 6 UK Cities get 5G mobile internet; Java applets formally deprecated.
    - 2020: Python 2 end of life.
- Technical lead of the *NewAthena* WFI Instrument Science Centre
  - 2038: *NewAthena* launches.

First warning: tech changes!

- Lead the UK *Swift* Science Data Centre.
  - Joined in 2006 as data centre scientist.
    - “Broadband” internet was a luxury (justified by on-call support).
    - Typical server: dual-core CPU, 4GB RAM. External RAID box for the archive.
    - “Flash” and Java applets were mainstream.
    - People thought IDL was a good idea.
      - (Though some people were trying out ‘Python’)
    - 2007: First iPhone goes on sale; xen incorporated into Linux kernel.
    - 2008: first Android phone goes on sale; Python 3.0 released.
    - 2012: 11 UK Cities get 4G mobile internet; automated storage tiering becomes mainstream.
    - 2013: Docker released. Astropy first release. Java applet support wanes.
    - 2014: Kubernetes released.
    - 2019: 6 UK Cities get 5G mobile internet; Java applets formally deprecated.
    - 2020: Python 2 end of life.
- Technical lead of the *NewAthena* WFI Instrument Science Centre
  - 2038: *NewAthena* launches.

2006

19 years

Now

13 years

2038

First warning: tech changes!

- Lead the UK *Swift* Science Data Centre.
  - Joined in 2006 as data centre scientist.
    - “Broadb
    - Typical
    - “Flash”
    - People
      - (Thou
    - 2007:
    - 2008:
    - 2012: ~~11 UK Cities get 4G mobile internet, automated storage tiering becomes mainstream.~~
    - 2013: Docker released. Astropy first release. Java applet support wanes.
    - 2014: Kubernetes released.
    - 2019: 6 UK Cities get 5G mobile internet; Java applets formally deprecated.
    - 2020: Python 2 end of life.
  - Technical lead of the *NewAthena* WFI Instrument Science Centre
    - 2038: *NewAthena* launches.

- **Design your *code* to be *infrastructure*-agnostic.**
- **Use modules/classes etc. for all infrastructure-related tasks.**
- e.g. so that changing from running a subprocess in a shell, to in a docker, to via K8s or via a batch processing system means changing one module, not 68 codes.

2006

years

Now

13 years

2038

Second warning: scope changes!

- On joining the UKSSDC, I was shown a grant proposal which included a pledge to write a tool to generate XRT light curves of GRBs…
- … so I did (Evans+ 2007, *An online repository of Swift/XRT light curves of GRBs* — A&A, 469, 379)

Second warning: scope changes!

- On joining the UKSSDC, I was shown a grant proposal which included a pledge to write a tool to generate XRT light curves of GRBs…
- … so I did (Evans+ 2007, *An online repository of Swift/XRT light curves of GRBs* — A&A, 469, 379)

Second warning: scope changes!

- On joining the UKSSDC, I was shown a grant proposal which included a pledge to write a tool to generate XRT light curves of GRBs…
- … so I did (Evans+ 2007, *An online repository of Swift/XRT light curves of GRBs* — A&A, 469, 379)
- … and then automated (and later expanded) Mike Goad's 'position enhancement'…

Second warning: scope changes!

- On joining the UKSSDC, I was shown a grant proposal which included a pledge to write a tool to generate XRT light curves of GRBs…
- … so I did (Evans+ 2007, *An online repository of Swift/XRT light curves of GRBs* — A&A, 469, 379)
- … and then automated (and later expanded) Mike Goad's 'position enhancement'…
- … and added automatic XRT spectra for GRBs.

Second warning: scope changes!

- On joining the UKSSDC, I was shown a grant proposal which included a pledge to write a tool to generate XRT light curves of GRBs…
- … so I did (Evans+ 2007, *An online repository of Swift/XRT light curves of GRBs* — A&A, 469, 379)
- … and then automated (and later expanded) Mike Goad's 'position enhancement'…
- … and added automatic XRT spectra for GRBs.
- "But what about those bursts where we don't know which source is the afterglow?"

Second warning: scope changes!

- On joining the UKSSDC, I was shown a grant proposal which included a pledge to write a tool to generate XRT light curves of GRBs…
- … so I did (Evans+ 2007, *An online repository of Swift/XRT light curves of GRBs* — A&A, 469, 379)
- … and then automated (and later expanded) Mike Goad's 'position enhancement'…
- … and added automatic XRT spectra for GRBs.
- "But what about those bursts where we don't know which source is the afterglow?"
- "It would be nice to apply these tools to the prompt data…"

Second warning: scope changes!

- On joining the UKSSDC, I was shown a grant proposal which included a pledge to write a tool to generate XRT light curves of GRBs…
- … so I did (Evans+ 2007, *An online repository of Swift/XRT light curves of GRBs* — A&A, 469, 379)
- … and then automated (and later expanded) Mike Goad's 'position enhancement'…
- … and added automatic XRT spectra for GRBs.
- "But what about those bursts where we don't know which source is the afterglow?"
- "It would be nice to apply these tools to the prompt data…"
- "Could people use these things for non-GRBs…? (2009)

Second warning: scope changes!

- On joining the UKSSDC, I was shown a grant proposal which included a pledge to write a tool to generate XRT light curves of GRBs…
- … so I did (Evans+ 2007, *An online repository of Swift/XRT light curves of GRBs* — A&A, 469, 379)
- … and then automated (and later expanded) Mike Goad's 'position enhancement'…
- … and added automatic XRT spectra for GRBs.
- "But what about those bursts where we don't know which source is the afterglow?"
- "It would be nice to apply these tools to the prompt data…"
- "Could people use these things for non-GRBs…? (2009)
  - Multi-messenger things started happening… (2011)

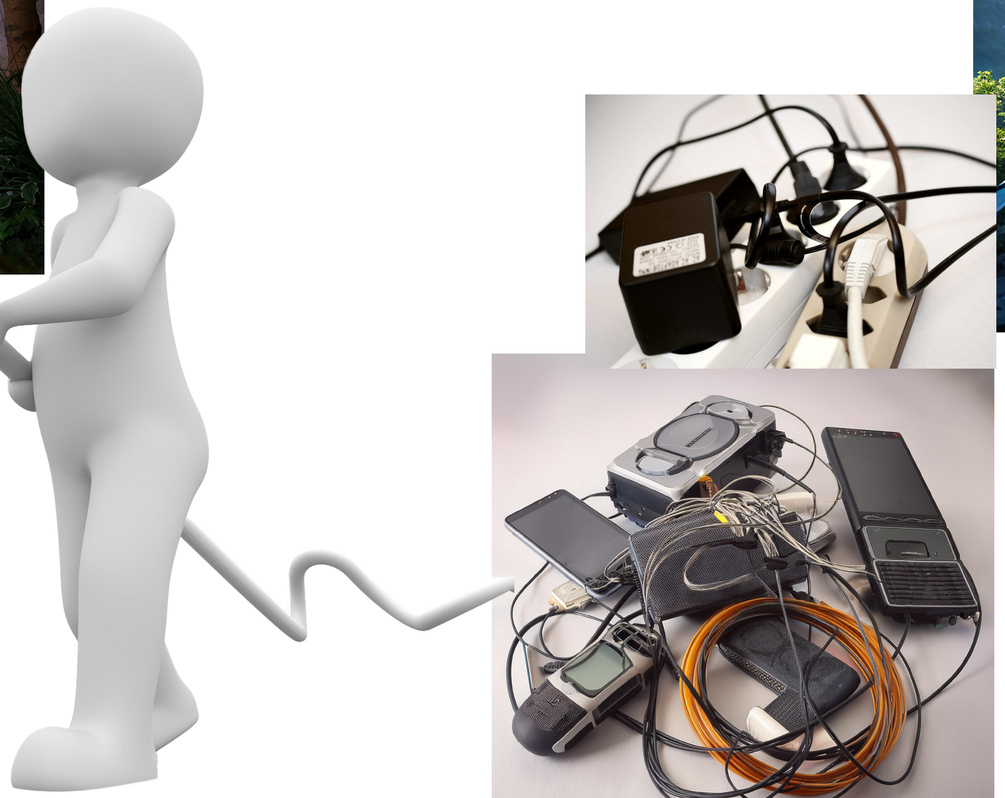
Unforeseen growth…

Unforeseen growth…

Unforeseen growth…

Unforeseen growth…

Unforeseen growth…



When the science landscape changes… adapt!

Science can change quickly — embrace the unforeseen.

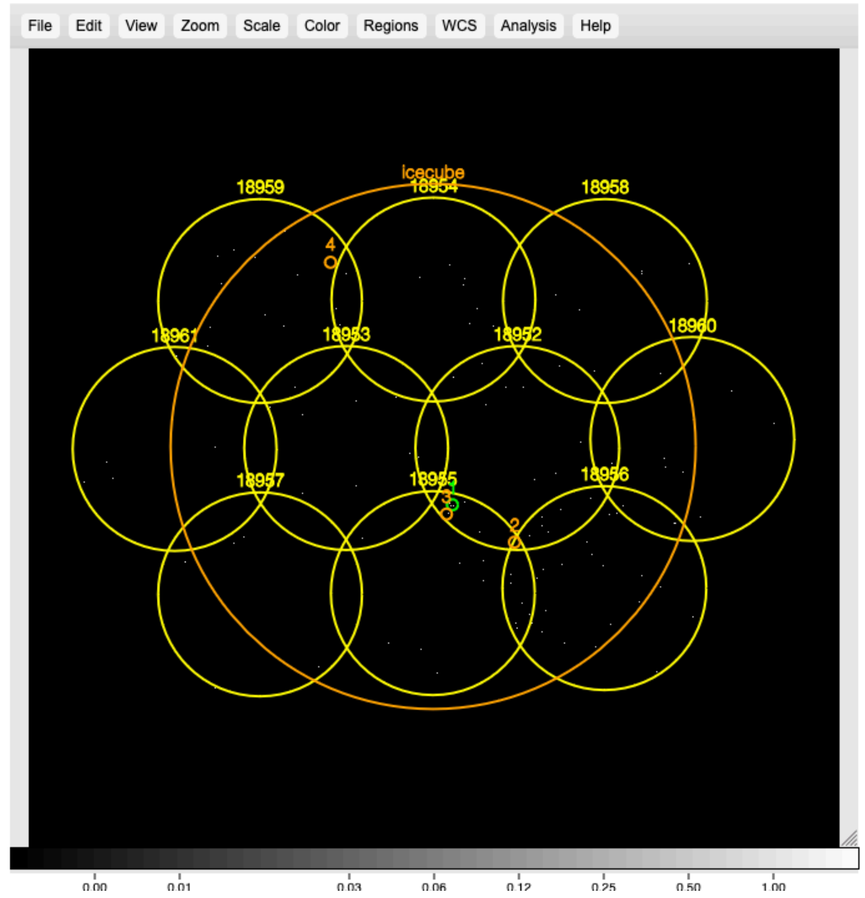
- For Swift, this started with IceCube neutrino doublets…
- … then HE neutrino triggers,
- … and ANTARES alerts.
- KM3NeT “tomorrow”.
- neutrinos (https://www.swift.ac.uk/neutrino/)

When the science landscape changes… adapt!

Science can change quickly — embrace the unforeseen.

- For Swift, this started with IceCube neutrino doublets…
- … then HE neutrino triggers,
- … and ANTARES alerts.
- KM3NeT “tomorrow”.
- neutrinos (https://www.swift.ac.uk/neutrino/)

Evans+ (2015); Adrián-Martínez+ (2016); Aartsen+ (2017); Keivani+ (2018)



So far **4** sources have been found in the XRT data. 3 sources match catalogued sources.

Coordinate type: Sexagesimal

#	RA (J2000)	Dec (J2000)	Err$_{90}$[1]	Detection Flag[2]	Rank[3]	Dist from IceCube pos	Exposure	Notes	Vizier
1 [details]	10h 56m 12.04s	+05° 15′ 41.3″	8.4′′ *	Good	4 (K)	7.5′	1.0 ks	Matches SIMBAD source: SDSS J105612.22+051544.6, 0.8″ away. Matches known X-ray source: 2SXPS J105612.0+051546 (from 2SXPS), 2.9″ away.	
2 [details]	10h 55m 42.04s	+05° 11′ 12.1″	7.0′′	Poor	3 (U)	15.2′	1.0 ks	Matches SIMBAD source: NVSS J105542+051112, 2.1″ away.	
3 [details]	10h 56m 15.13s	+05° 14′ 37.7″	8.2′′	Poor	3 (U)	8.3′	1.0 ks		
								Matches SIMBAD source: PM	

When the science landscape changes… adapt!

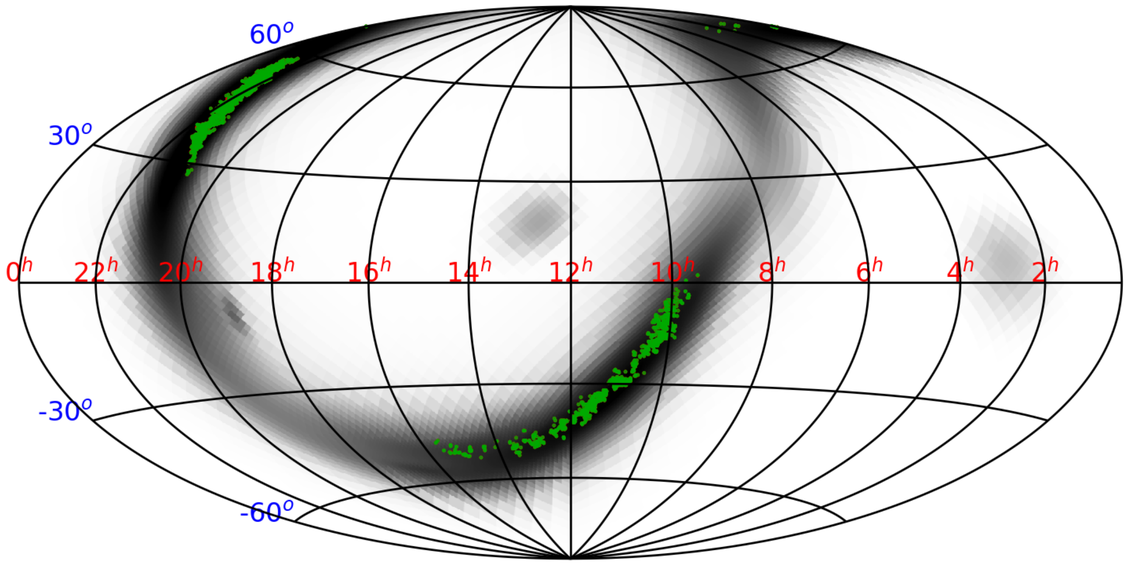
Science can change quickly — embrace the unforeseen.

- For Swift, this started with IceCube neutrino doublets…
- … and FRBs and then of course, Gravitational Waves.
  - (https://www.swift.ac.uk/LVC/)

When the science landscape changes… adapt!

Science can change quickly — embrace the unforeseen.

- For Swift, this started with IceCube neutrino doublets…
- … and FRBs and then of course, Gravitational Waves.
  - (https://www.swift.ac.uk/LVC/)

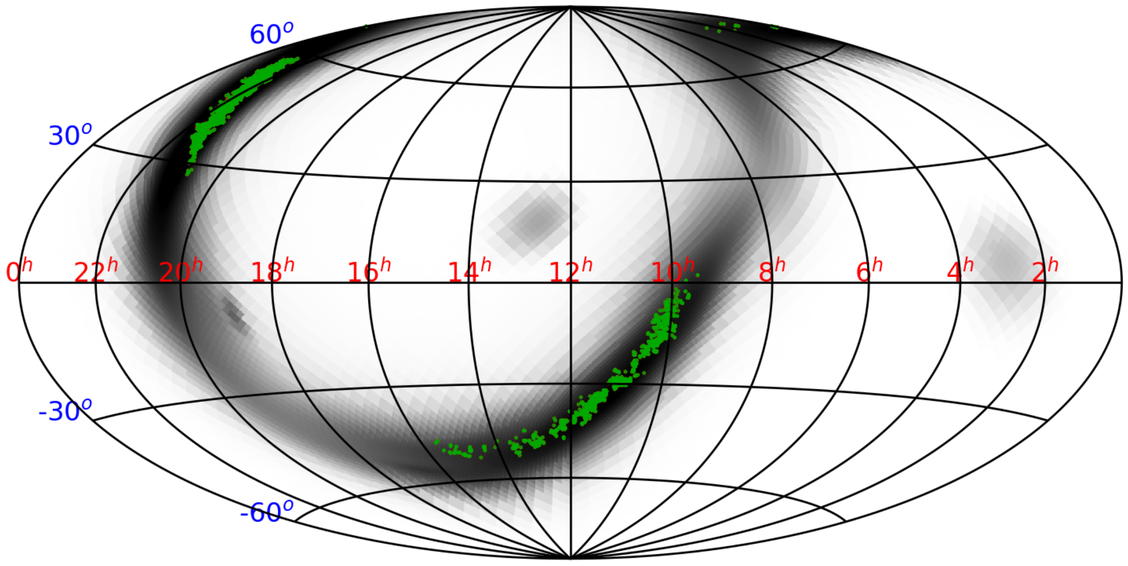


GW details	
Trigger Date:	2017-02-27 at 18:57:31.000 UT
Trigger type:	Compact Binary Coalescence.
Distance:	193 ± 61 Mpc
$P_{ContainsNeutronStar}$:	1
$P_{EMBright}$:	1
False alarm rate:	1.426e-7 Hz = 1/81 d
GraceDB:	GraceDB
P_{BAT} (orig map)	0.111299 (raw) 0.085083 (convolved)
XRT details	
First observation:	T_0 + 264.7 min
Fields planned:	1414 [Search]
Fields observed:	1408
TargetID range:	7006400—7007830
LVC Prob observed	0.089 (raw), 0.23 (conv)
LVC Prob to observe	7.11e-4 (raw), 1.35e-3 (conv) [LALInference]
XRT sources:	59
Candidate afterglows:	0
Possible afterglows:	3
Unlikely afterglows:	16
Known X-ray sources:	40

When the science landscape changes… adapt!

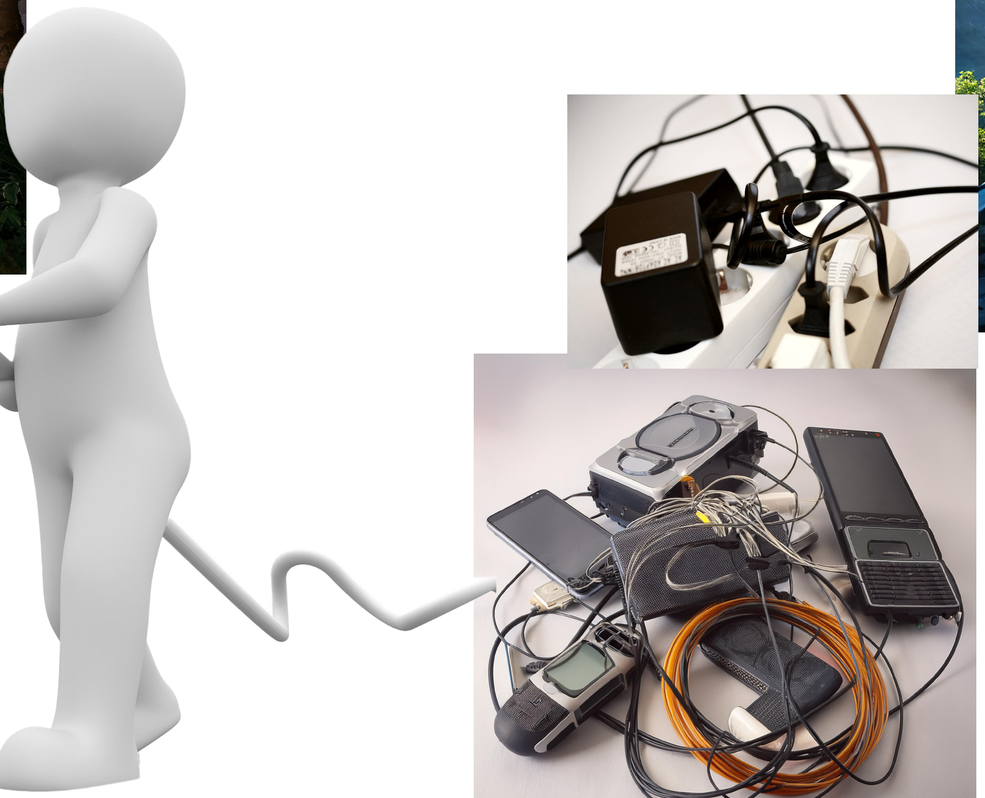
Science can change quickly — embrace the unforeseen.

- For Swift, this started with IceCube neutrino doublets…
- … and FRBs and then of course, Gravitational Waves.
  - (https://www.swift.ac.uk/LVC/)



GW details	
Trigger Date:	2017-02-27 at 18:57:31.000 UT
Trigger type:	Compact Binary Coalescence.
Distance:	193 ± 61 Mpc
$P_{ContainsNeutronStar}$:	1
$P_{EMBright}$:	1
False alarm rate:	1.426e-7 Hz = 1/81 d
GraceDB:	GraceDB
P_{BAT} (orig map)	0.111299 (raw) 0.085083 (convolved)
XRT details	
First observation:	T_0 + 264.7 min
Fields planned:	1414 [Search]
Fields observed:	1408
TargetID range:	7006400—7007830
LVC Prob observed	0.089 (raw), 0.23 (conv)
LVC Prob to observe	7.11e-4 (raw), 1.35e-3 (conv) [LALInference]
XRT sources:	59
Candidate afterglows:	0
Possible afterglows:	3
Unlikely afterglows:	16
Known X-ray sources:	40

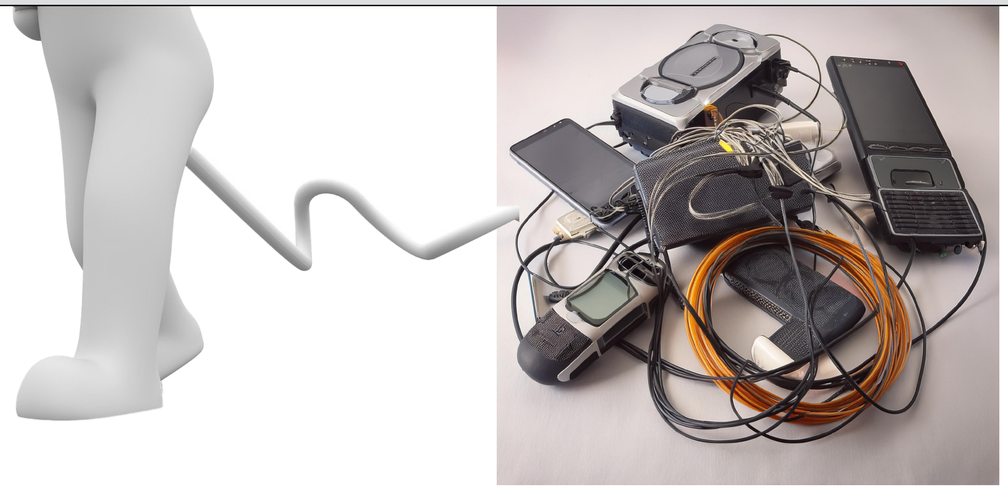
“Oh yay! I make some better tools”



“Oh yay! I make some better tools”

“Oh yay! I make some better tools”

- **Write good developer-facing documentation from the start, all the time.**
- **Build and maintain a full test harness.**
- “What happens if I delete this probably obsolete code?”



If only I could start again…

- Lead the UK *Swift* Science Data Centre.
- Technical lead of the *NewAthena* WFI Instrument Science Centre.

If only I could start again…

- Lead the UK *Swift* Science Data Centre.
- Technical lead of the *NewAthena* WFI Instrument Science Centre.

What have I learned?

If only I could start again…

- Lead the UK *Swift* Science Data Centre.
- Technical lead of the *NewAthena* WFI Instrument Science Centre.

What have I learned?

- **Design your *code* to be *infrastructure*-agnostic.**
- **Use modules/classes etc. for all infrastructure-related tasks.**
- e.g. so that changing from running a subprocess in a shell, to in a docker, to via K8s or via a batch processing system means changing one module, not 68 codes.

If only I could start again…

- Lead the UK *Swift* Science Data Centre.
- Technical lead of the *NewAthena* WFI Instrument Science Centre.

What have I learned?

- **Design your *code* to be *infrastructure*-agnostic.**
- **Use modules/classes etc. for all infrastructure-related tasks.**
  - e.g. so that changing from
    via a batch processing sy
- **Write good developer-facing documentation from the start, all the time.**
- **Build and maintain a full test harness.**
  - “What happens if I delete this probably obsolete code?”

The starting point: what is the goal?

The starting point: what is the goal?

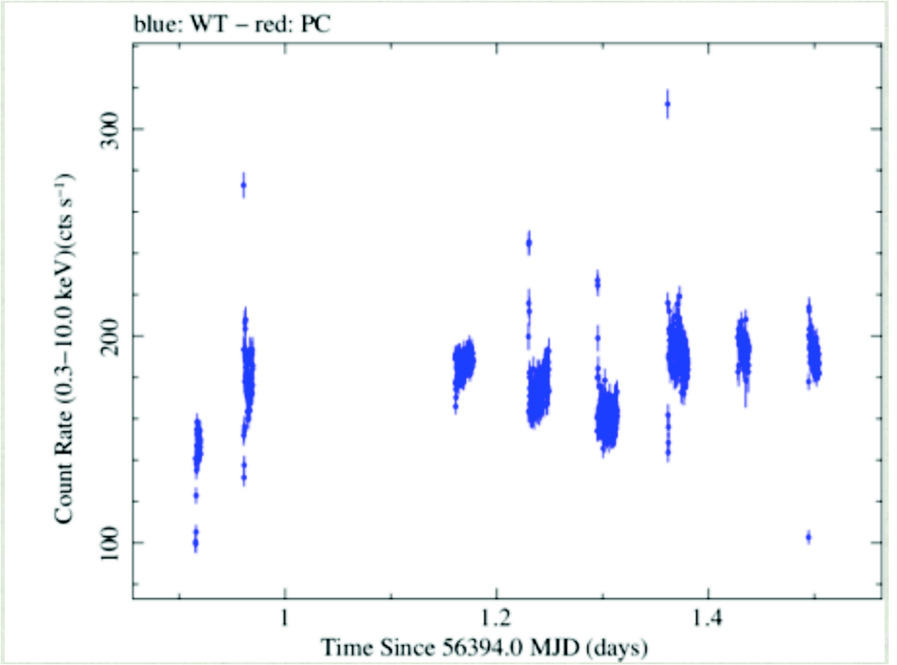
- Maximising the science return from the project.

The starting point: what is the goal?

- Maximising the science return from the project.
  - **Creating software / systems that allow scientists to focus on science.**

The starting point: what is the goal?

- Maximising the science return from the project.
  - **Creating software / systems that allow scientists to focus on science.**
- Data analysis can be complex, intricate and difficult.
- It can require pretty esoteric knowledge.
- Errors are NOT always easy to spot.
- No-one wants bad analysis → bad science to be published.



We should make the bar to using our data as low as possible — **but not zero?**

How do we achieve this?

- User focus! (I know, users are annoying…)
- "Full stack" dev-scientist is increasingly unfeasible…
  - …but development by users is advantageous…
    (Ever tried inserting cross references into a Word document?)
  - …and can someone please explain that to our funders?
- Agile approaches (user stories, continually working product, regular user checkins) … but are they all agile?
- A user community regularly interacting with the developers is important.
- What users want ≠ what we think they want…
- … and ≠ what each other wants.

Speaking of users…

Speaking of users…

- They will find a critical bug within (typically) $\hbar/2$ s of a new release.

Speaking of users…

- They will find a critical bug within (typically) $\hbar/2$ s of a new release.
- They always want more…

Harold Copping

Speaking of users…

- They will find a critical bug within (typically) $\hbar/2$ s of a new release.
- They always want more…
  - You don’t have to provide them all! And certainly not all now.

Harold Copping

Speaking of users…

- They will find a critical bug within (typically) $\hbar/2$ s of a new release.
- They always want more…
  - You don’t have to provide them all! And certainly not all now.
  - But some are great things to add — triage is key.

Harold Copping

The problem with users

- They will find a critical bug within (typically) 3.2 ns of a new release.
- They always want more…
  - You don’t have to provide them all! And certainly not all now.
  - But some are great things to add — triage is key.

Harold Copping

The problem with users

- They will find a critical bug within (typically) 3.2 ns of a new release.
- They always want more…
  - You don’t have to provide them all! And certainly not all now.
  - But some are great things to add — triage is key.
- You will spend *hours* crafting beautiful user documentation.

Harold Copping

The problem with users

- They will find a critical bug within (typically) 3.2 ns of a new release.
- They always want more…
  - You don’t have to provide them all! And certainly not all now.
  - But some are great things to add — triage is key.
- You will spend *hours* crafting beautiful user documentation.
  - DO ask a trusted colleague to proof-read it.

Harold Copping

The problem with users

- They will find a critical bug within (typically) 3.2 ns of a new release.
- They always want more…
  - You don’t have to provide them all! And certainly not all now.
  - But some are great things to add — triage is key.
- You will spend *hours* crafting beautiful user documentation.
  - DO ask a trusted colleague to proof-read it.
  - DO NOT ask your boss to check it’s OK 😅.

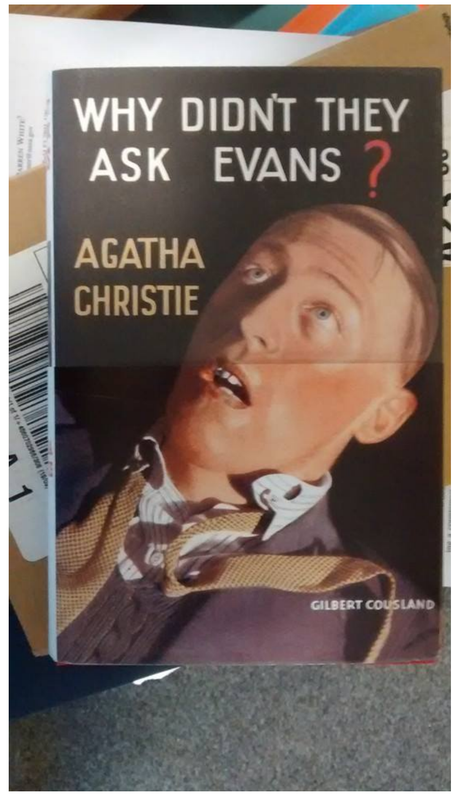
Harold Copping

The problem with users

- They will find a critical bug within (typically) 3.2 ns of a new rele
- They always want more…
  - You don't have to provide them all! And certainly not all now.
  - But some are great things to add — triage is key.
- You will spend *hours* crafting beautiful user documentation.
  - DO ask a trusted colleague to proof-read it.
  - DO NOT ask your boss to check it's OK 😅.
  - Your users will **never** read it 😱😭.

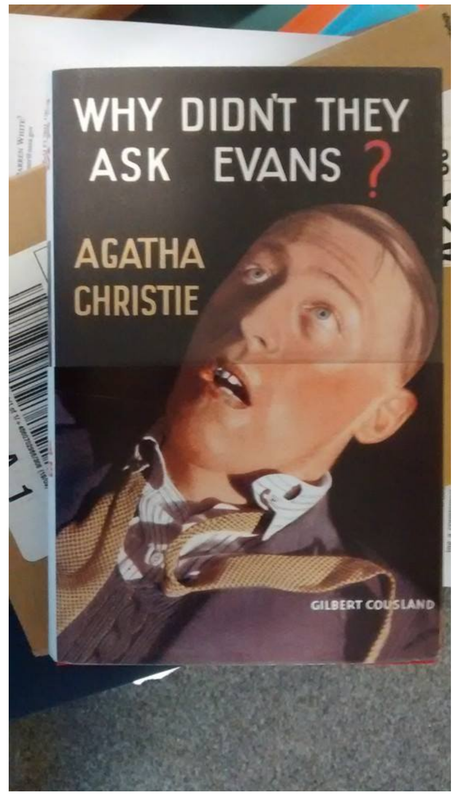
The problem with users

- They will find a critical bug within (typically) 3.2 ns of a new release.
- They always want more…
  - You don’t have to provide them all! And certainly not all now.
  - But some are great things to add — triage is key.
- You will spend *hours* crafting beautiful user documentation.
  - DO ask a trusted colleague to proof-read it.
  - DO NOT ask your boss to check it’s OK 😅.
  - Your users will **never** read it 😱😭.
- Users can be shockingly rude (and wrong), or wonderful.



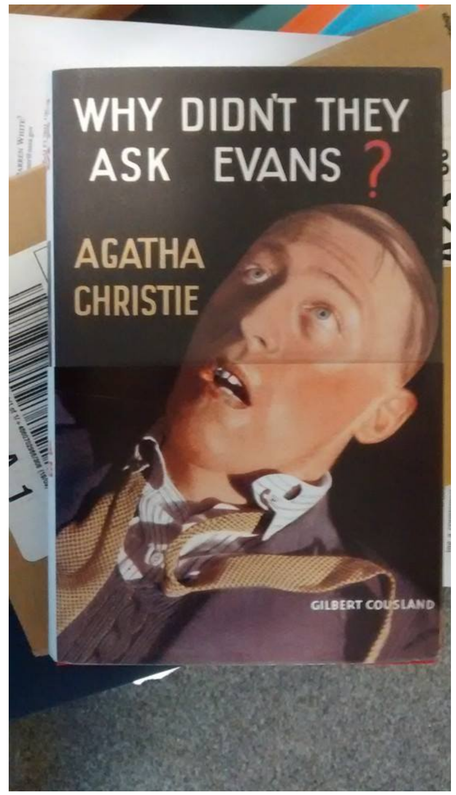
The problem with users

- They will find a critical bug within (typically) 3.2 ns of a new release.
- They always want more…
  - You don't have to provide them all! And certainly not all now.
  - But some are great things to add — triage is key.
- You will spend *hours* crafting beautiful user documentation.
  - DO ask a trusted colleague to proof-read it.
  - DO NOT ask your boss to check it's OK 😅.
  - Your users will **never** read it 😱😭.
- Users can be shockingly rude (and wrong), or wonderful.
  - Keep a stressball to hand before opening emails and never reply immediately 😝.



The problem with users

- They will find a critical bug within (typically) 3.2 ns of a new release.
- They always want more…
  - You don't have to provide them all! And certainly not all now.
  - But some are great things to add — triage is key.
- You will spend *hours* crafting beautiful user documentation.
  - DO ask a trusted colleague to proof-read it.
  - DO NOT ask your boss to check it's OK 😅.
  - Your users will **never** read it 😱😭.
- Users can be shockingly rude (and wrong), or wonderful.
  - Keep a stressball to hand before opening emails and never reply immediately 😝.
  - You are not a universal PhD supervisor / non-coauthor collaborator.



Factoring this into our approach

Build your garden shed assuming it will one day power the world.

Assume:

- The scope will widen, more than you expect — and then some.
- The technology it uses will change.
- You will not be the last person in your role.
- Users will do something “stupid”.

Factoring this into our approach

Build your garden shed assuming it will one day power the world.

Assume:

- The scope will widen, more than you expect — and then some.
  - “Modular” design.
  - Developer-facing docs (auto docs?).
  - Test harness.
  - Test harness.
- The technology it uses will change.
- You will not be the last person in your role.
- Users will do something “stupid”.

Factoring this into our approach

Build your garden shed assuming it will one day power the world.

Assume:

- The scope will widen, more than you expect — and then some.
  - “Modular” design.
  - Developer-facing docs (auto docs?).
  - Test harness.
  - Test harness.
- The technology it uses will change.
- You will not be the last person in your role.
- Users will do something “stupid”.

Factoring this into our approach

Build your garden shed assuming it will one day power the world.

Assume:

- The scope will widen, more than you expect — and then some.
- The technology it uses will change.
  - Infrastructure agnosticism (separate interfaces from implementation where possible).
  - Developer-facing docs (auto docs?).
    - Design map (heterogeneous dependency and call graph).
  - Test harness.
  - Employ a sysadmin.
  - Factor “learning new stuff” in your schedule.
- You will not be the last person in your role.
- Users will do something “stupid”.

Factoring this into our approach

Build your garden shed assuming it will one day power the world.

Assume:

- The scope will widen, more than you expect — and then some.
- The technology it uses will change.
- You will not be the last person in your role.
  - Did I mention developer-facing docs?
  - `git commit` messages matter*.
  - “No siloed knowledge”.
- Users will do something “stupid”.

* *“Work, you steaming pile of elephant dung”* is, apparently, “not a helpful record.”

Factoring this into our approach

Build your garden shed assuming it will one day power the world.

Assume:

- The scope will widen, more than you expect — and then some.
- The technology it uses will change.
- You will not be the last person in your role.
- Users will do something “stupid”*.
  - Build a user community to test, feedback etc. — know what your users need.
  - Lowering the bar to data use comes with drawbacks:
    - *“A common mistake that people make when trying to design something completely foolproof is to underestimate the ingenuity of complete fools.”* — Douglas Adams
    - Warning messages are helpful… and unhelpful.
    - You are responsible for the tools, users are responsible for their use of them.

** “Stupid”: anything I didn’t expect.*

And last

- Coding is fun, science is fun; try to keep it that way.
- Force-feeding rude users their own feet is illegal in most countries*.
  - “Accidentally” adding their IP to `/etc/httpd/conf.d/blacklist.conf` isn’t*. 😜
- Time pressures are real: skimping docs / design etc. costs a LOT more than it saves.
- Astrophysical software is fundamental to allowing any project to fulfil its goals.
  - It is not a thankless task, honest!

* Legal disclaimer: I am not a lawyer, any legal opinions should be treated as “completely made up but sound like they should be true.”