CRM05

Too Much of a Good Thing

UCM Streams and Flood Control

Adonica Gieger

CM Consultant, Adonica Ltd.

adonica@adonica.co.uk

IBM Rational Software Development Conference 2008

WHERE TEAMS ARE R-HEROES

IBM

Rational. software

CRM05: UCM Streams and Flood Control

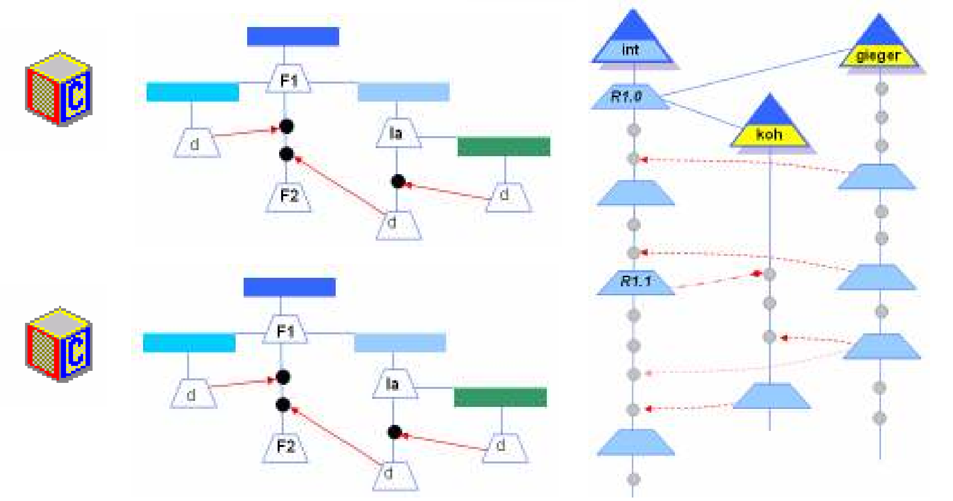
- Streams in the UCM context
- Common models
  - Stumbling stairs
  - Mainline
  - Adapted stairs
- Generic release streams
  - Initial
  - Delta 1
  - Delta 2
  - Variants
- Whiteboard Q&A

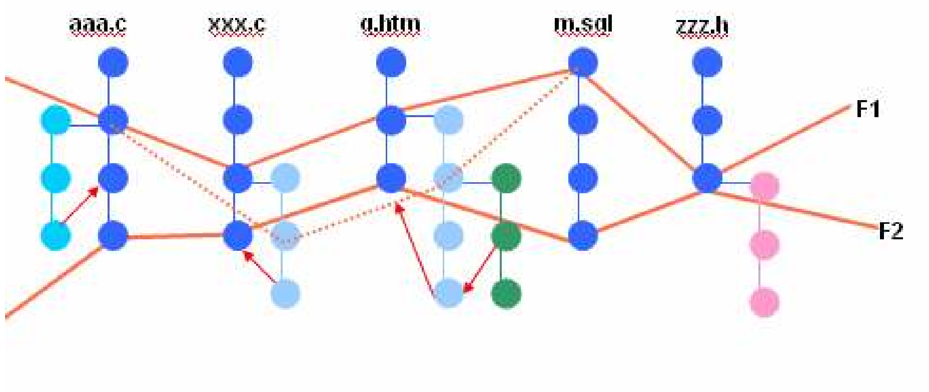
- Additional explanations appear in PowerPoint‘s Notes mode

Stream

- UCM object introduced in 1998
- Guards a like-named base CC branch type
- Maps to time-line objects in components, which hold baselines

- Each project has at least one stream, its integration stream
- No limit on number or depth of streams
- Each stream has a ...
  - One display parent
  - One or more effective parents
  - One default delivery target

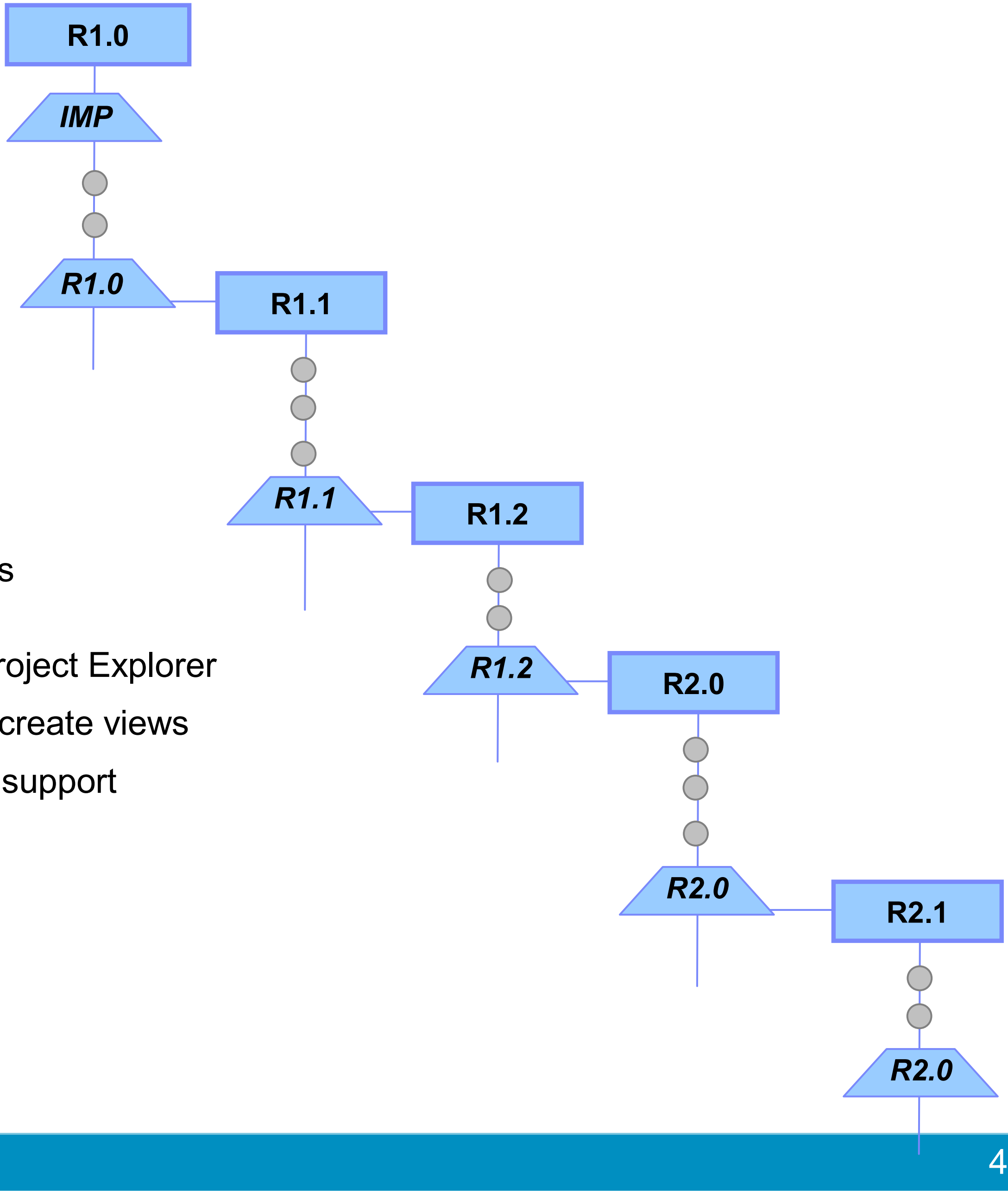




Stumbling Stairs

- Project per release
- Chronological order

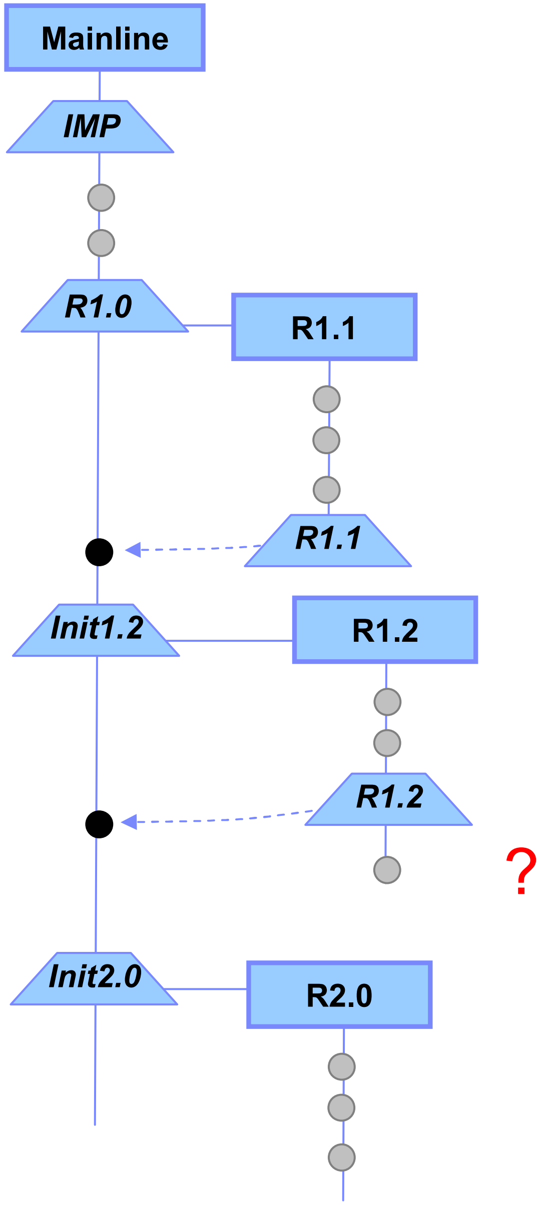
- Rapidly too deep
- Awkward to view in version tree and Component Browser
- Long extended paths create problems in Windows and GUI
- Plethora of streams/projects slows Project Explorer
- Developers must change projects, recreate views
- Awkward for multiple live releases in support or long-horizon enhancements



Mainline

- Project per release
- Chronological order
- Deliver to reference stream after release

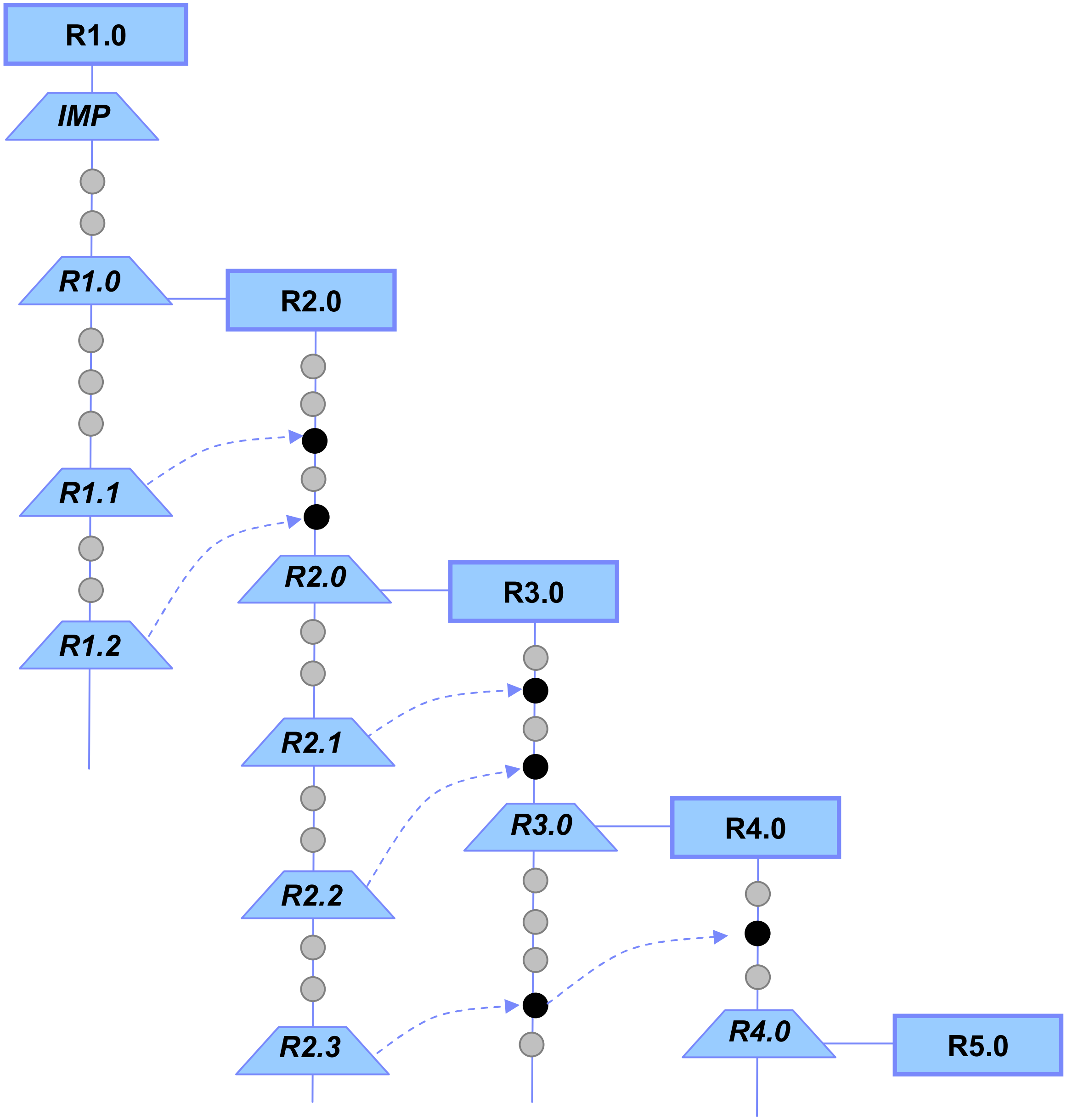
- Difficult to maintain in order
- Plethora of streams/projects slows Project Explorer
- Developers must change projects, recreate views
- Awkward for multiple live releases or long-horizon enhancements



Adapted Stairs

- Project per major release
- Chronological order
- Re-used for minor releases

- Slowed appearance of problems but did not eliminate them



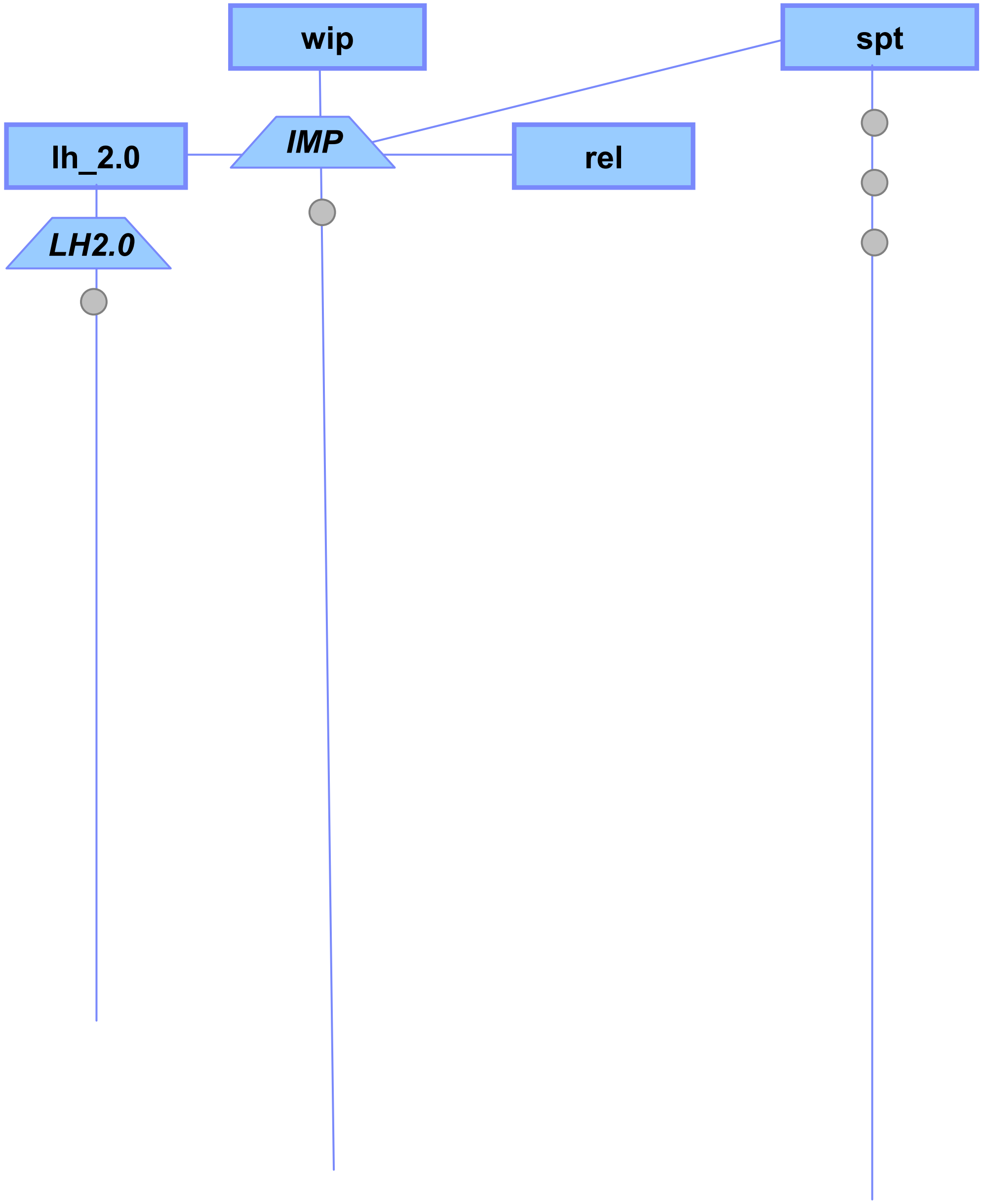
Root Problem?

- Project per release – why?
- Chronological order – why?
- What is a release?

- Development continues. Why shouldn‘t the project?

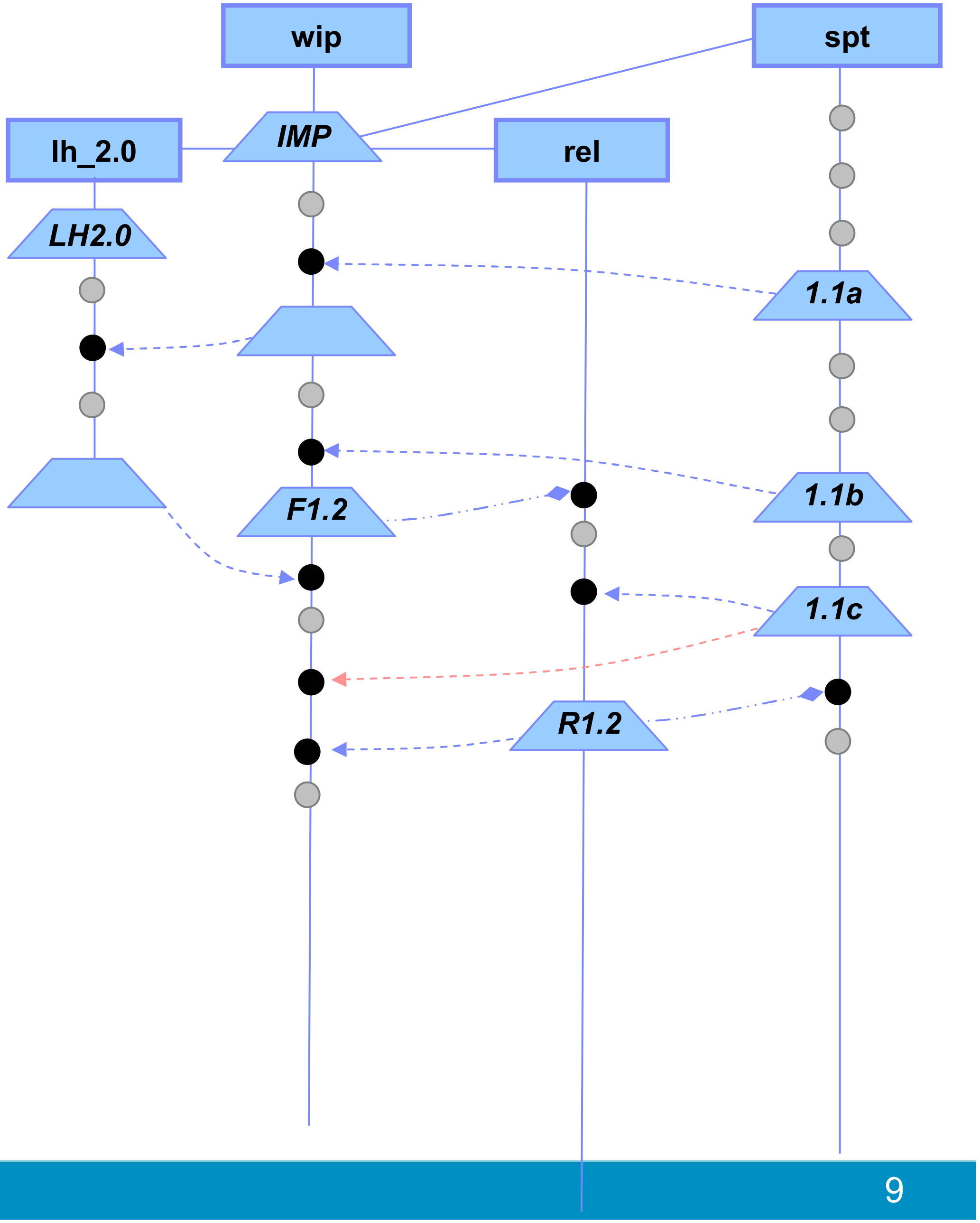
Initial

- Create wip
  - Import current live (1.1), baseline as IMP
- Start projects at baseline IMP
  - Create rel
  - Create spt
  - Create lh_2.0
  - Import work to date on 2.0, baseline as LH2.0
- Work begins …



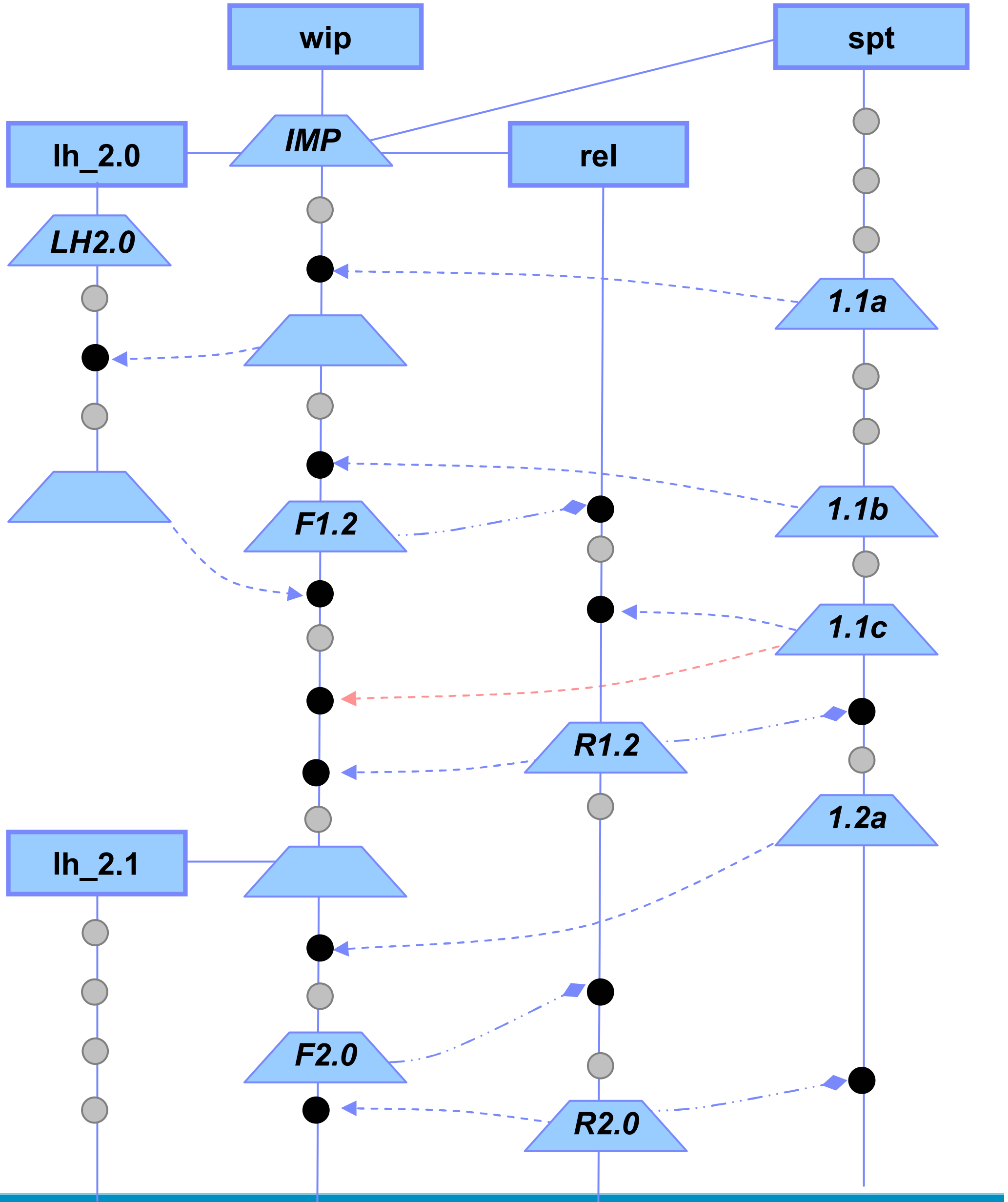
Delta 1

- Release 1.1a bugfix
  - Deliver to wip
- Deliver baselines on wip to lh_2.0
- Work continues…
- Release 1.1b bugfix
- F1.2 begins freeze of release 1.2
  - Rebase rel, deliver lh_2.0 to wip
- Work continues…
- Release 1.1c bugfix
  - Deliver to rel & wip (optional)
- Release R1.2
  - Rebase spt, deliver to wip



Delta 2

- Work continues…
- Begin lh_2.1 from wip baseline
- Release 1.2a bugfix
  - Deliver to wip
- Work continues…
- F2.0 begins freeze of release 2.0
  - Rebase rel
- Work continues…
- Release R2.0
  - Rebase spt, deliver to wip



Variants and Drawbacks

- GRS assumes only one live release in support at any one time
  - To support multiple live releases use one stream per live release
  - Naming convention important to avoid confusion

Thanks!!

- Questions?
- Suggestions?
- Derision?

DINNER SPONSORED BY:

REAL
SOLUTIONS