

# Changes in Java release process

- releases history
- planned release schema
- Fedora/RHEL distribution alignment
- AdoptOpenJDK

jvanek@redhat.com  
OpenJDK QA engineering  
Devconf 25.1.2018

<https://jvanek.fedorapeople.org/devconf/2018/changesInjavaReleaseProcess.pdf#lastSlide>

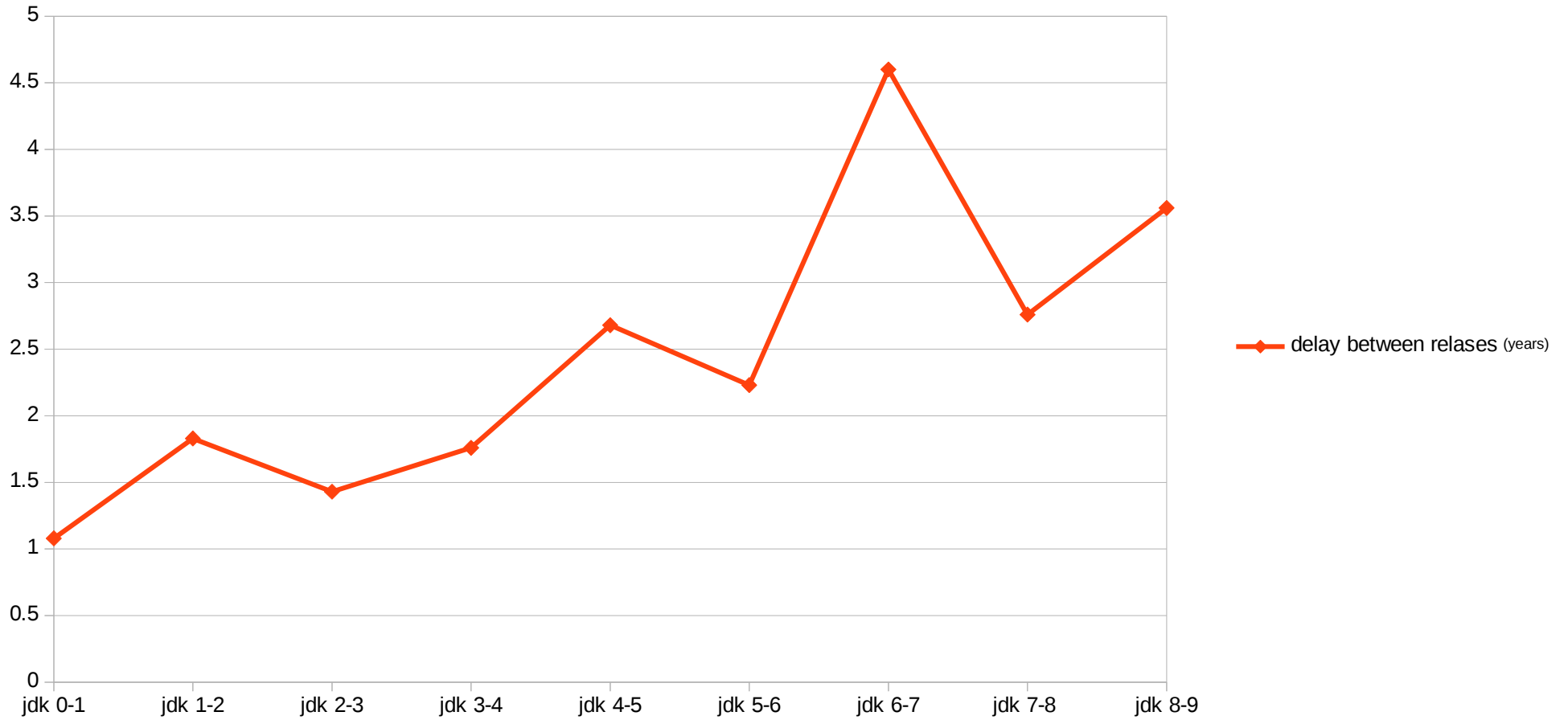
# Changes in Java release process

- Shorter presentation
  - Need feedback for distribution alignment at the end
- Os
  - Win?
  - Fedora?
  - RHEL?
  - Other
- Java for development
  - 7? 8? 9?

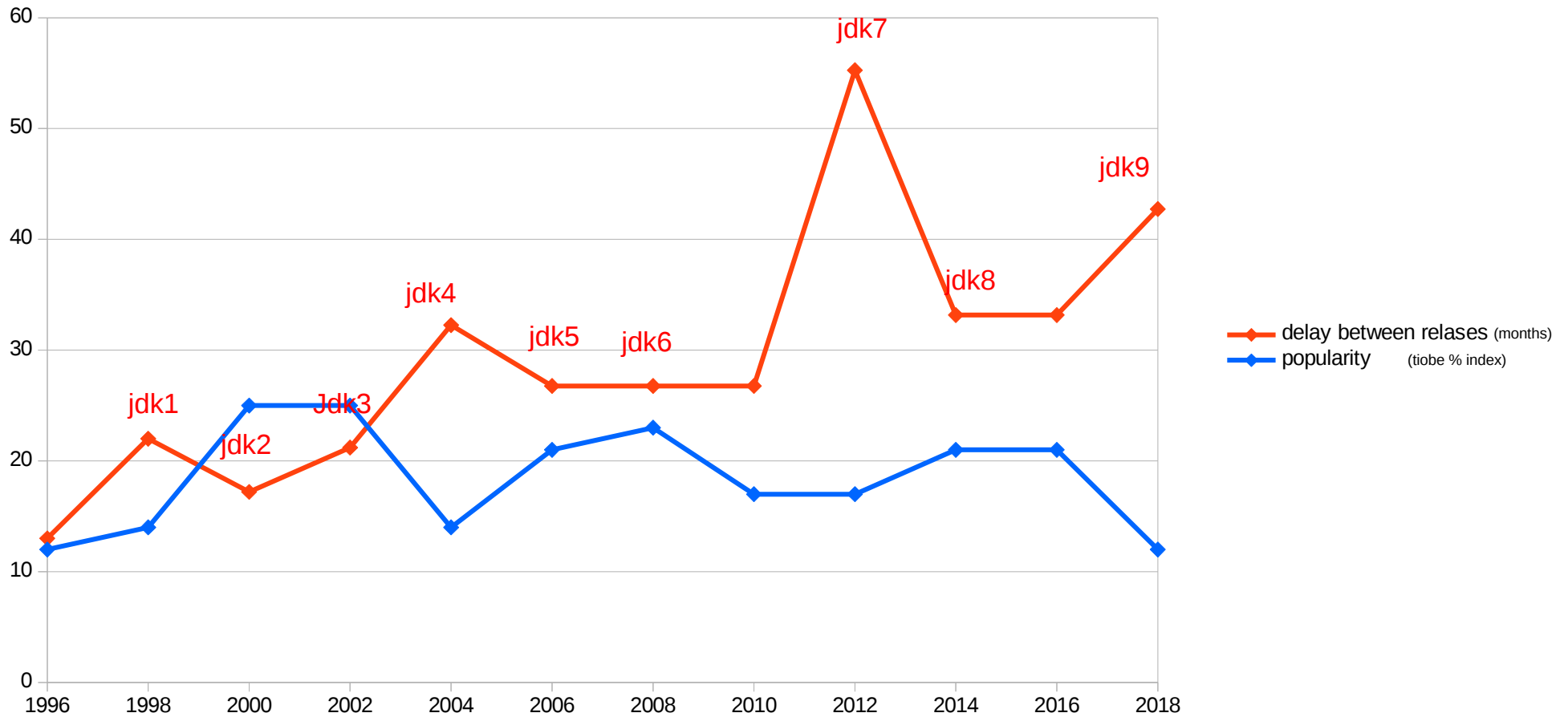
# Releases history

- JDK 1.0 (January 23, 1996)
- JDK 1.1 (February 19, 1997)
- 2SE 1.2 (December 8, 1998)
- J2SE 1.3 (May 8, 2000)
- J2SE 1.4 (February 6, 2002)
- J2SE 5.0 (September 30, 2004)
- Java SE 6 (December 11, 2006)
  - 2010 - Oracle acquired Sun
- Java SE 7 (July 28, 2011)
- Java SE 8 (March 18, 2014)
- Java SE 9 (September 21, 2017)
  - All were feature-driven

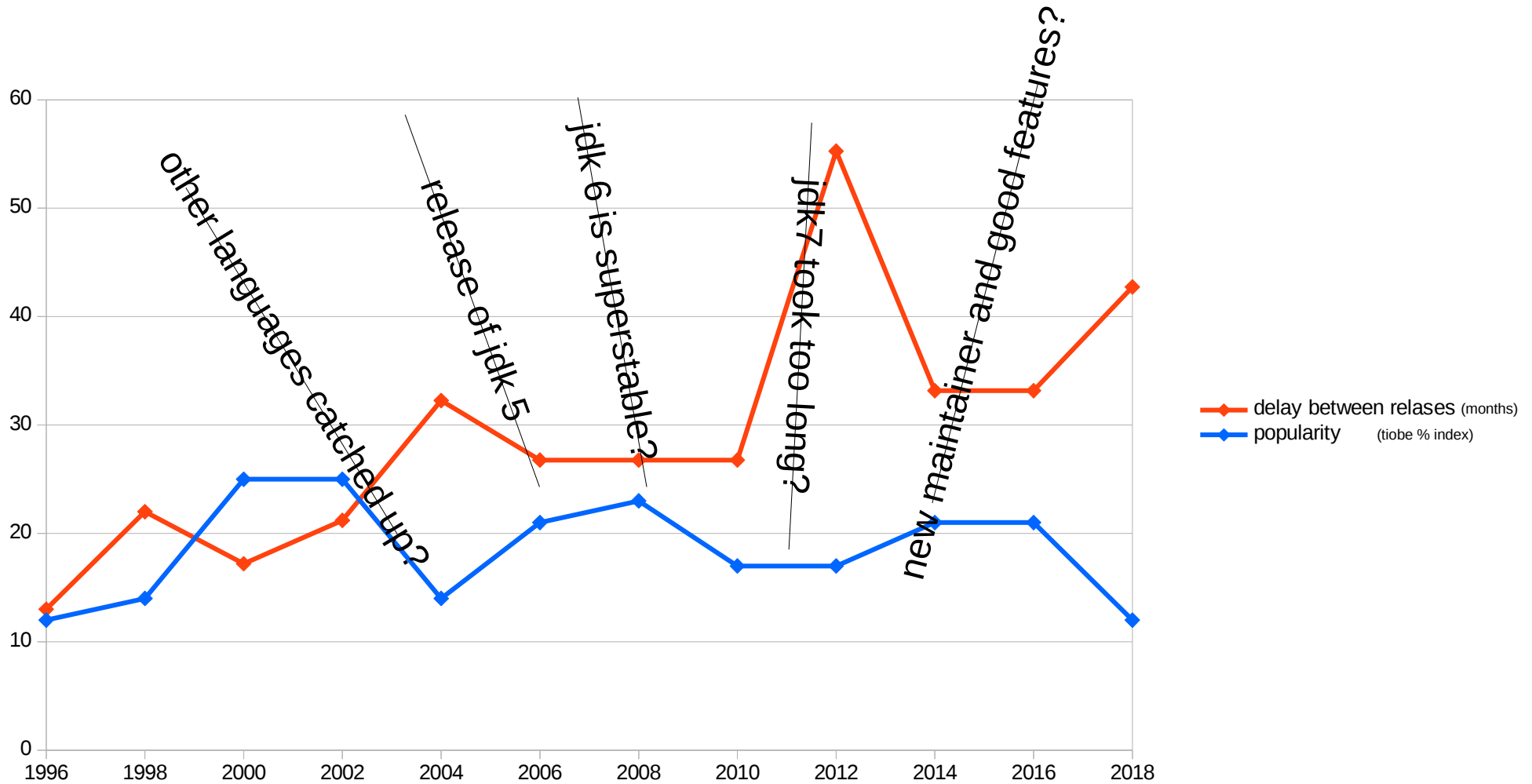
# Releases history



# Releases history



# Releases history



# Oracle faded release plan

- Java SE 7 (July 28, 2011)
  - 2 years are too short for business, but ok for new features
  - 3 years + LTS should be ok for business
  - big new features in separate repos
    - merging into main forest when “ok”
- Java 8 (March 18, 2014)
  - took 2.5 years
  - already too long for nice features
- Java 9 (September 21, 2017)
  - took 3.5 years
    - really too much for new features
    - testing of new features is clumsy

# One more look into recent history

- Java SE 6 (12.11, 2006)
    - 2010 - Oracle acquired Sun
  - Java SE 7 (7.28, 2011)
  - Java 8 (3.18, 2014)
  - Java 9 (9.21, 2017)
    - can be considered (imho) modular JDK public beta 1
      - jdk10 - 3/2018 public beta 2
      - jdk11 - 9/2018 true modular jdk?
- 2008 started Project Jigsaw
  - pretty soon abandoned for JDK7
  - 7/2012 even discarded out of JDK8
  - 3/2014 merged into main forest
  - 7/2017 claimed to be completed

...however...

I know how it looks, but now it is done.  
Good or bad, it was necessary step



# New release plan

- Issues in previous lifecycle which would not be repeated
  - forest merged with feature in semi completed state
    - JDK9 was delayed several times because of (merged) Jigsaw
  - main release waiting for project
  - projects are hard to test
  - isolation of projects not exactly healthy
  - developers need much faster cycles
  - business need much longer (even independent) cycles



# New release plan



- no more feature driven development
- new release every 6 months
- new LTS (Oracle's) every two years
  - Anybody can proclaim anything as LTS and support it
    - Depending on usage, Red Hat may choose to take over an LTS release after Oracle publicly EOLs it
- projects merged when done
  - like really done
- projects still in separate forests
  - the only hope here lies in AdoptOpenJDK
  - So the issue with merged buggy finished project remains. (?)
- after JDK 11 (first LTS, September 2018), Oracle will opensource the remaining (possible) parts of “Oracle JDK” and will start to ship OpenJDK
  - they may ship it as OracleJDK with some support condi
  - all burden goes to OpenJDK and AdoptOpenJDK
- date based version given up for continuous line
  - ufff
  - Date version (eg. 18.3) kept internally

```
$ java -version
openjdk version "10-internal" 2018-03-20
OpenJDK Runtime Environment (build 10-internal+0-host)
OpenJDK 64-Bit Zero VM (build 10-internal+0-host, interpreted mode)
```



# Short-Term-Support updates

- Rare
  - It is easy to push change to future jdk
  - But to backport to STS...
- <http://hg.openjdk.java.net/jdk-updates>
  - Project for STS updates (currently jdk9)
- Hopefully backports to LTS will go better
  - But experience says otherwise
- A lot of work for distributions
  - Or AdoptOpenJDK for users of static builds

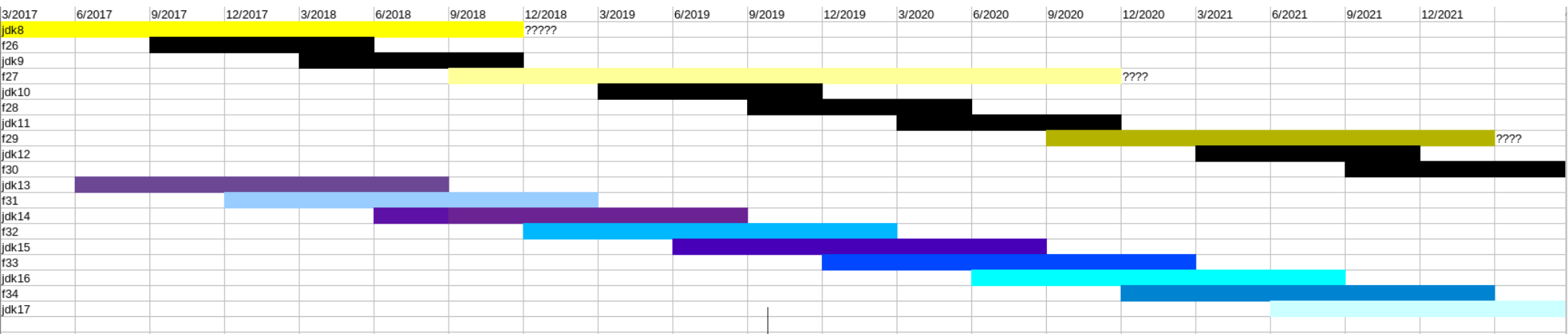
# Fedora downstream – work in progress!

- Fedora claims to be the newest technologies distro
- Fedora have approx 6 months, time driven release cycle
- No intersection with planned 6 months STS and occasional LTS
  - Each LTS will be shipped
- To pass fedora review for each new jdk would be overkill
  - Most (we hope all) STS will be shipped
    - Will keep java-openjdk as rolling release for STS
    - Will pack new java-X-openjdk for each LTS
    - Occasional retirement of java-openjdk to allow major changes
    - Change main JDK (used in buildroot) will be system-wide change and will be aligned with Fedora release cycle
      - If it will ever be STS one is to be decided (if so, how?)
      - Partially different integration
      - How to proceed with updates STS jdk → STS jdk in Lifetime of fedora?
        - **Bytecode compatibility** (will change at STS?)

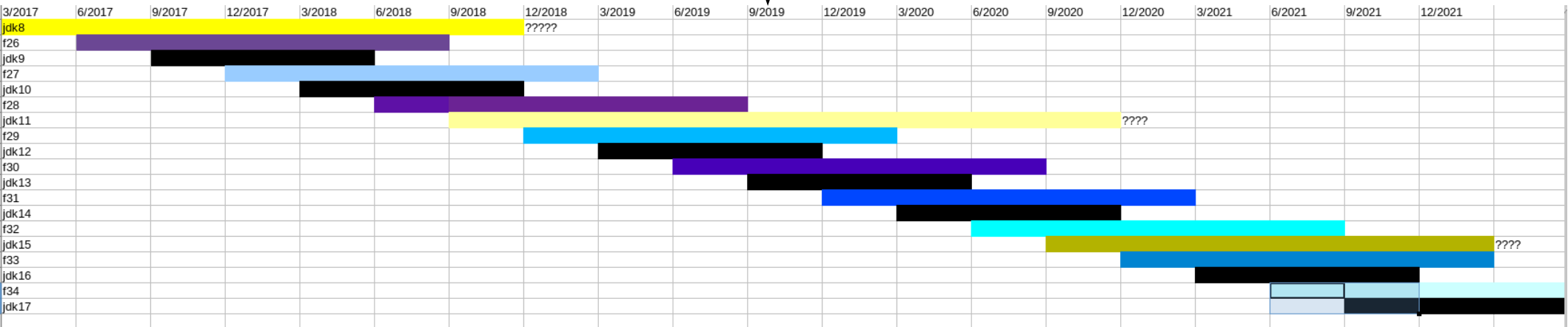
# Fedora downstream – work in progress!

- F8...F15...F17...  
...
  - F21 12/2014  
...
  - F26 06/2017
  - F27 12/2017
  - F28 06/2018
- JDK6 → 7 as techpreview → 7 → ...
  - JDK8 release 3/2014
    - F19+20 techpreview
  - Java-1.8.0-openjdk package
    - Made main JDK in F21
  - JDK9 released 9/2017 (changed to STS soon after release)
    - As tech preview in copr
    - Since F26 as official tech-preview
    - Although providing java, still tech-preview
  - JDK10 planned 3/2018 STS
    - Will remain techpreview (public beta?-)
  - JDK11 planned 9/2018 LTS
    - ?

# Fedora downstream – work in progress!

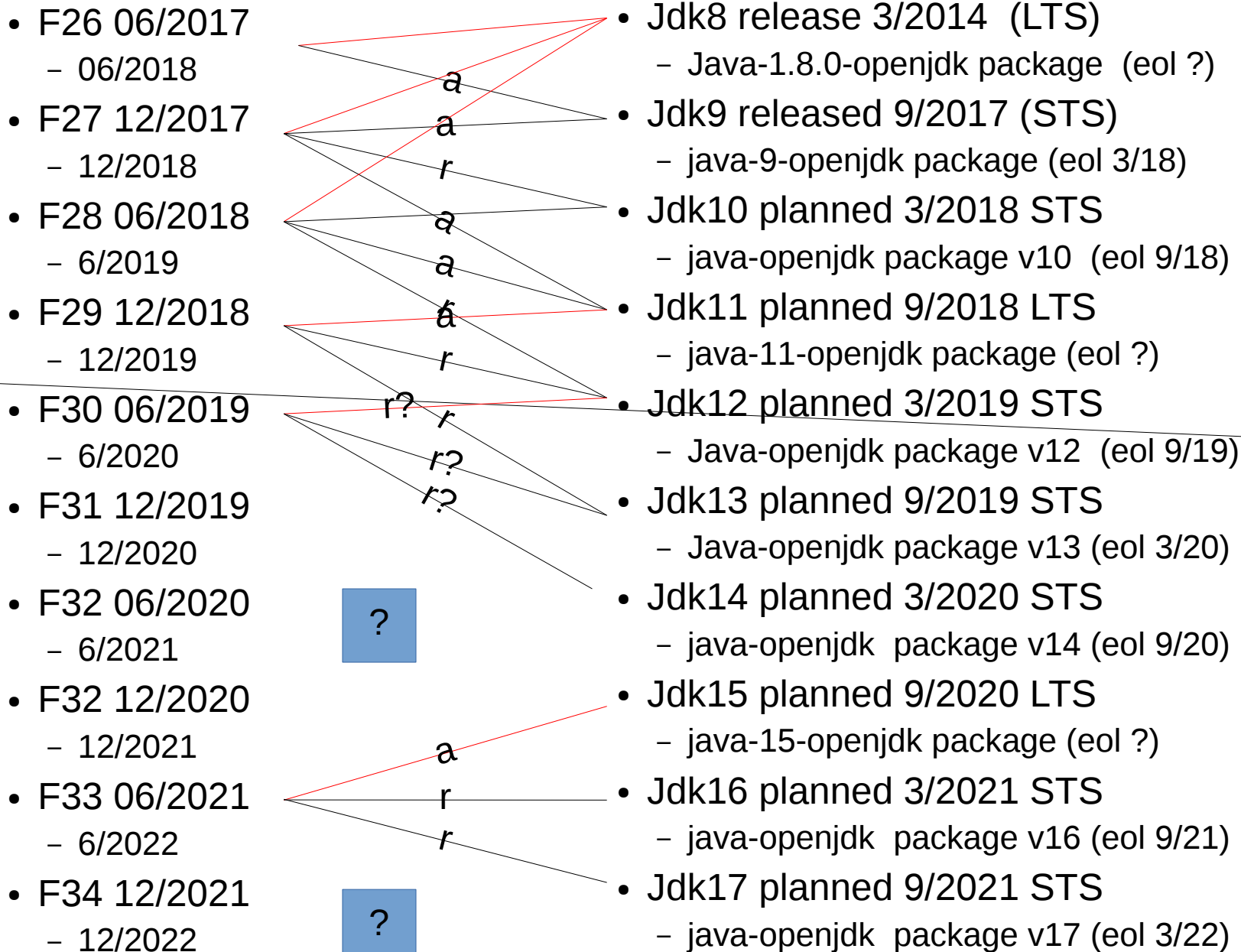


Sort ↓



# Fedora downstream – work in progress!

main jdk and tech preview, Adding or Replacing



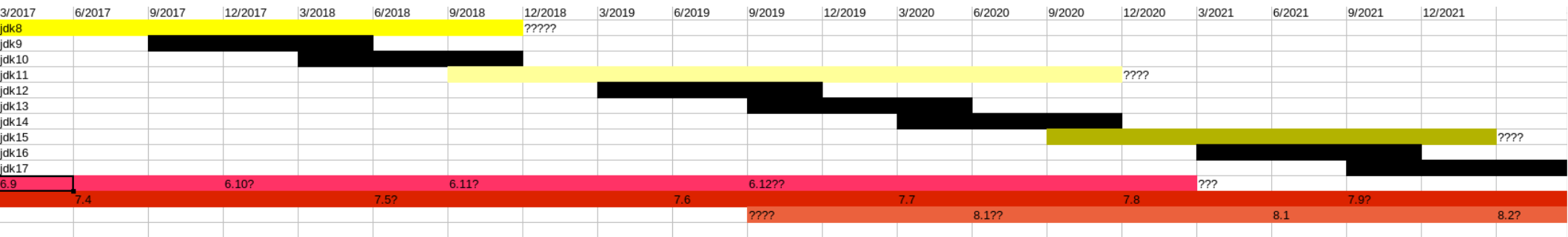
Another jigsaw is probably not possible

# Fedora downstream – work in progress!

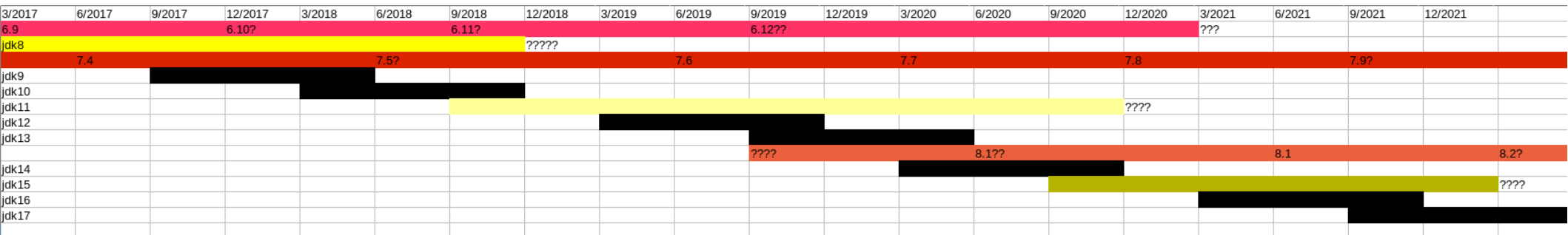
- Having rolling release of java-openjdk with occasionally added java-X-openjdk in the given timeframe have several issues:
  - Rolling release will probably enforce removal of similar-parallel JDKs
  - If so, STSs will not be allowed to install in parallel
    - As each STS will replace previous one in java-openjdk, STS will not be able to coexists
    - Occasional retirement of java-openjdk to adapt most recent changes of java-x-openjdk integration maybe necessary
  - Using STS as main JDK for fedora maybe necessary, should be possible, but hard to judge all consequences
  - LTSs overlap, and will be able to coexist, but Fedora lifetime is too short to use it
  - Hints and ideas welcome
    - IRC: Freenode/#fedora-java
    - IRC: OFTC/#openjdk
    - Email: distro-pkg-dev@openjdk.java.net



# RHEL downstream – work in progress!



Sort ↓



\* "RHEL 6.10 and higher, 7.5 and higher and RHEL 8 release dates are written only for orientation and are not based on any real data.

# RHEL downstream – work in progress!

- RHEL is stable distribution
  - Still is trying to get new packages in
- 10+ years lifecycle is longer than any regular OpenJDK
- Bytecode compatibility will be held
  - Even if build JDK is EOLed, it will get javac/javadoc security fixes
  - Main JDK can upgrade to a higher version if necessary
- More fun is with Y updates
- It is not exactly easy to put new package to RHEL
  - Each LTS will be attempted to be shipped

# RHEL downstream – work in progress!

- RHEL6
  - Build JDK remains OpenJDK6
  - Main JDK is JDK7
    - Replaced JDK6 in 6.5
  - JDK8 available
- RHEL7
  - Build JDK remains OpenJDK7
  - JDK7 and 8 are now equal
  - JDK9 not available, JDK10 probably same
  - JDK11 should be available
    - Decision is still pending
- RHEL8
  - Guessing!
  - "Main JDK will either be 8 or 11
  - All LTS-es will likely be available

# AdoptOpenJDK

- Public infrastructure to provide builds of OpenJDK
  - Currently terribly non-deterministic
    - portable\* build from f27 will not work on RHEL6
    - And there is not guaranteed host right now
  - However support for all possible architectures
  - Hope to support all project forests
    - Can make future features accessible like a charm

\* portable build is as “static” build of JDK as possible. Such build must be done against oldest possible libraries, and then it works fine on same or newer.

# AdoptOpenJDK

- Many big players (Amazon, Alibaba...)
  - Have patched JDKs
    - But nothing back to upstream
    - Oracle is unhappy about that
- Maintaining both oracle JDK and OpenJDK is costly
  - Opening Oracle JDK and start to deliver OpenJDK is logical step
  - Drop custom build infrastructure and use AdoptOpenJDK may be another one

# Questions, Feedback, Ideas....

- Announcement of new JDK release process
  - <http://mail.openjdk.java.net/pipermail/discuss/2017-September/004281.html>
- Blog-post about the new release process
  - <https://mreinhold.org/blog/forward-faster>
- History of project jigsaw
  - <http://openjdk.java.net/projects/jigsaw/>
- Announcement of delay of jigsaw to JDK9
  - <https://mreinhold.org/blog/late-for-the-train>
- Latest versioning (accepted) proposal
  - <http://openjdk.java.net/jeps/322>
- Tiobe java popularity
  - <https://www.tiobe.com/tiobe-index/java/>
- Fedora X (7,15,22,26...) changes and schedule
  - <https://fedoraproject.org/wiki/Releases/X/Schedule>
  - <https://fedoraproject.org/wiki/Releases/X/ChangeSet>
- History of JDK
  - [https://en.wikipedia.org/wiki/Java\\_%28programming\\_language%29#Versions](https://en.wikipedia.org/wiki/Java_%28programming_language%29#Versions)