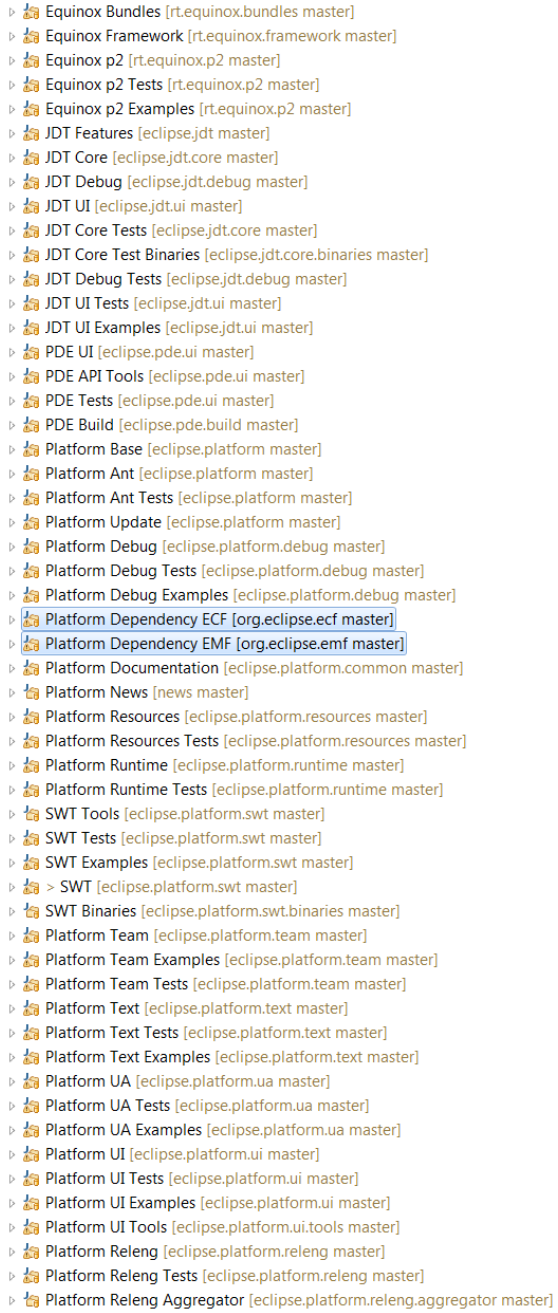
The Traditional Eclipse Technology Stack

Where are we heading?

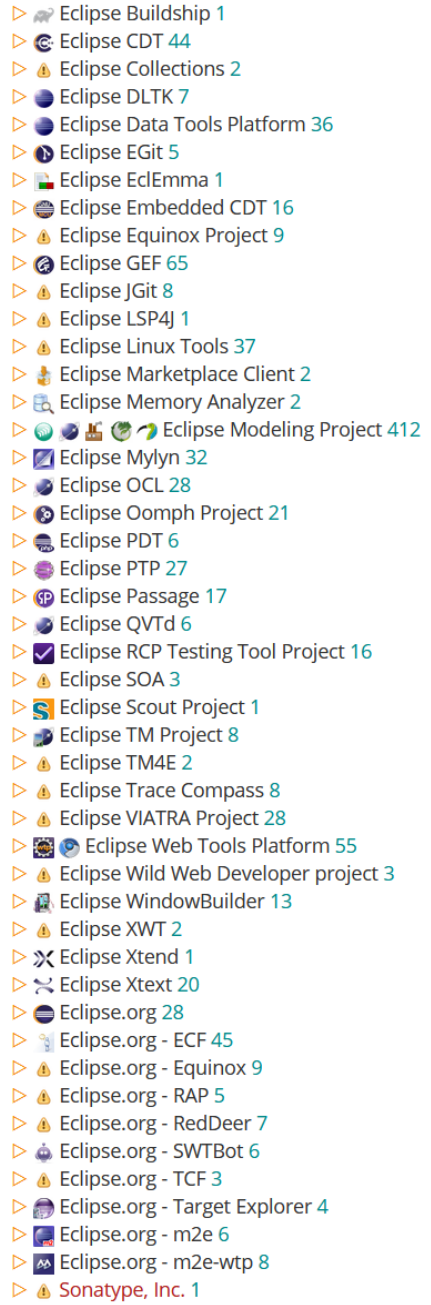
The Rich Client Platform Stack Basis

- RCP Platform + Equinox +
  - Eclipse Modeling Framework (for e4)
  - Eclipse Communications Framework for (p2)
  - Jetty (for help)
- Additional “Platform” Projects
  - JDT
  - PDE
  - Team
- Key deliverables
  - Eclipse project p2 repository
  - Eclipse SDK packages



The Full IDE Technology Stack

- The > 70 SimRel projects
  - https://download.eclipse.org/oomph/archive/simrel/
- Key deliverables
  - The release repository, composes
    - The SimRel train repository
    - The EPP repository
  - The EPP packages
  - The Eclipse Installers
    - Updated catalogs based
      - the release repositories
      - the Eclipse project repositories



Problems and Risks

- Project starvation and extinction
  - Damages reputation: Eclipse is not a reliable source of technology
- Poor quality deliverables
  - Blocks adoption
  - Damages reputation: Eclipse is buggy
- Contributing factors
  - Poor quality tracking, poor testing, and poor reporting
    - It’s too much work, so it doesn’t get done properly, and isn’t automated
  - Frequent full releases
    - Helps to deliver new features quickly and reduces develop overhead, but…
    - Contributes to disruptive changes and regressions in every release, moving the cost downstream
  - Unstable “APIs” in up stream components create work for downstream components
    - Many things are not possible using purely APIs
    - Lack of respect for consumers
      - If you don’t like it, come and fix it yourself
      - Open source principles are used as a weapon