Ideas for YP 4.1

Brainstorming for the 4.1 release. Ideas can be added and elaborated, people can indicate interest or ownership. Ultimate aim is to translate these into bugzilla entries which take over from this document.

Proposed Milestone Dates:

Milestone	Build Date	Release Date
YP 4.1 M1	**2022-05-30**	**2022-06-10**
YP 4.1 M2	**2022-07-11**	**2022-07-22**
YP 4.1 M3	**2022-08-22**	**2022-09-02**
YP 4.1 M4	**2022-10-03**	**2022-10-28**
YP 3.1.16	**2022-04-25**	**2022-05-06**
YP 3.1.17	**2022-06-06**	**2022-06-17**
YP 3.1.18	**2022-07-18**	**2022-07-29**
YP 3.1.19	**2022-08-29**	**2022-09-09**
YP 3.1.20	**2022-10-10**	**2022-10-21**
YP 3.4.4	**2022-05-02**	**2022-05-13**
YP 4.0.1	**2022-05-16**	**2022-05-27**
YP 4.0.2	**2022-06-27**	**2022-07-08**
YP 4.0.3	**2022-08-08**	**2022-08-19**
YP 4.0.4	**2022-09-19**	**2022-09-30**
YP 4.0.5	**2022-10-31**	**2022-11-11**

Key:

- {} has final owner
- [] contact person who understands item (not a commitment to doing it)
- () person who submitted item or comments from the person indicated

People:

- RP - Richard Purdie
- RB - Ross Burton
- MH - Michael Halstead
- AK - Armin Kuster
- Jon - Jon Mason
- TO - Tim Orling

Planned items for 4.1:

-

Ideas pulled from YP 3.5 discussion:

- Git fetcher alternates support (PaulG)
- Improve gathering/reporting on https://www.yoctoproject.org/reproducible-build-results/
- Document Hash Equivalence (MichaelO)
- SystemReady Support to meta-arm (Ross/Jon)
- Switchable toolchain [Ross]
    - Easily switch between gcc, clang, icc without each layer needing to reimplement the logic.
- Clang in core (akuster)
- Yocto Compatibility (akuster)
    - Needs better definition. Like do dependent layers need to pass? How to run, what to run, approval process #13285
    - Would like to add a self-test for this to make it consistent {AK}
        - Patches rejected.
- License Improvements:
    - Use debug source information to compute license field information for comparison? (RP)
    - Generate SPDX manifest output with each deploy artefact (RP)
        - Is this done? (akuster)
- resulttool improvements:
    - Allow generation of html reports/graphs (RP) #13330
- Project Dashboard?
- Revise autobuilder test matrix [RP]
    - Do we test all kernel versions? All tiny? All -rt?
        - Add RT to *-alt ?? (armpit)
    - Meta-cgl [Armpit]?
    - Meta-security - declined by TSC [[Armpit]
        - Found alt solution
- Expand new testing manual
- Make meta-gplv2 unnecessary (by creating a configuration that disables gpl3 dependencies or replaces them with maintained non-gpl3 alternatives). Mbition is looking into this, as people realize meta-gpl2 is not a long term solution. (AlexK)
- Add toaster to getting started docs?
- Investigate recursive variable dependency resolution with a view to improving parsing performance (RP/Joshua?)
- Replacing multiprocessing in Cooker Parsing with something simpler (Joshua)

Security:

- Create dedicated security team (RP)

    - Become CNA? (https://cve.mitre.org/cve/cna.html)
        - This only makes sense is we need to assign our own #
    - PSiRT??
- Security/CVEs (needs subitems creating)
    - Proactive (AK)
    - Reactive (AK)
        - There is an increase of patches being sent to the list already
- Monitoring (AK)
- Policy (AK)
    - Do we fix all or just those above a particular score
- Enable memory resident bitbake by default
    - Needs to work on autobuilds, current very broken
    - Fix a number of YP bugs on this topic

Other ideas which could be developed:

- Patchtest/patchwork (needs subitems creating)
- Layer setup tooling for the core (needs investigation/discussion and subitems creating)

Ideas in Bugzilla with Owner:

- Revise release artifacts {RP} - Bug #11791
- Switch to memres bitbake by default? {RP} - Bug #13023
- Containers: Agree on what belongs in oe-core vs other layers {Bruce}
    - See Bugzilla: #13051, #13052, #13053
    - There are two main areas to consider for this: development host / testing and target execution. The bugzilla entries will be split into these broad categories, since we may not have time to do both.
    - oe-core needs to only get the minimal plumbing/capabilities, since there is no one size fits all solution to this, and many of the existing tools are far to heavy / complex for oe-core to carry (20-30 significant native and target depends/rdepends) .. this is due to far too broad (for what the build process needs) in the tools.
    - In fact, what oe-core needs is somewhat custom, since none of the existing tools are capable of quickly creating a simple OCI “image”.
    - The final question will be about compatibility, whether we want to produce docker registry compatible images (so they could immediately be pushed to a registry) or if we want to produce OCI compat images and then leave the conversion to docker as an ecosystem question. Bugzilla entries will be created for both.

- Migrate from pkgutil style namespace packages to native namespace packages {Chris Larson} - Bug #13034
    - https://packaging.python.org/guides/packaging-namespace-packages/#native-namespace-packages
- Testopia removal/bugzilla upgrade { MH } #13015

    - We are not using it for QA anymore (AK)
- SDK improvements:
    - Switch SDK to be self extracting Python script {Joshua Watt} #13259
    - Multiconfig SDK {Alejandro} #5421
- Enable installable image in wic/systemd-boot (Tim Orling) #13025
- Move bbclass python code to lib/oe (parsing performance/mem footprint)
- FIgure out way to avoid ptest dependency creep
- Enable more Release process in the AutoBuilder
- Package Maintainers document guidelines
- Improve autobuilder task definition/reporting