Error messages are not actionable Facilities to debug sbt are limited Warnings to educate users should be discouraged Lack of colors in sbt output Unclear, cryptic error messages Display of compilation errors could be improved and errors could be appropriately numbered **Usability** Get sbt 1.0 as soon as possible Absence of `value` produces a runtime error instead of a compile error Single jar download of sbt 'value' semantics are difficult and yield unexpected No support for offline mode results when used in normal Scala code situations Use NIO for everything (if, while, by-name, inside anon functions). Consider adding 'help' methods for concise One-line description of tasks and settings are not Add MiMa by default on sbt explanations of kevs **New features** enouah Source dependencies (they already exist, but they Publishing artifacts could be easier are not fully functional) It's difficult to know how to change sbt to do things Add sbt-pgp by default on sbt for which you don't have a formulaic solution for Test harness for sbt to check reproducible behaviour Reproducible builds Documentation needs to improve Dependency lock file Which scope should I use? Sbt docs are not regenerated when changes occur Scoping is unclear and unintuitive -- there should be What scope is being used? to the repository better documentation and conceptual simplification. How does scope delegation work? Division between 0.13 and 1.0 docs is confusing Learning experience / sbt model It's not clear why lazy is used to define setting and Documentation is not up to date in comparison with task keys. previous sbt versions Community feedback Syntax is too complicated **Documentation** No discoverability for plugins (Scaladex integration) on sbt Lack of strongly opinionated way of writing builds Interaction between Scala files in project and sbt Port SO questions to documentation Missing documentation on all the macros and use of Configurations are difficult to use reflection Documentation is written by experts, neglecting beginners that are not familiar with the fundamentals The state of the source code makes it hard for An empty or missing build.sbt is a valid project people to contribute (should be fixed as it is in scala-extras) Improving scripted, sbt's test framework, would Bare keys are not scoped per-build but per-project make writing plugins and hacking on sbt faster Sbt test framework is slow and tricky No way to define a root project that auto-aggregates Documentation is poor and it's not official **Current misbehaviours** (as it does when there is no root project) Contributing experience Maintainers are too conservative and cautious (for The distinction between 'Global' and 'ThisBuild' is business reasons) and that slows downs the open not clear source community Releases happen rarely -- speeding up release specific dependencies cadence is a must Explore adding Bill of Materials support Dependency management Ivy is slow Cours Missing support for fine-grained builds Missing support for incremental compilation Zinc, the incremental compiler within jars Slow in 0.13.x, fixed in 1.0.x Reloads are slow Scala instances are leaked and classpaths are not Consumes too much memory cached -- can be fixed **Performance** Watching sources in big projects is slow, seems to affect `testOnly` too

Much faster

Reduces code duplication for Windows