ESIP Software Guidance Part 2

Soren Scott

ESIP Summer Meeting July 2016

From Software Assessment Part 1…

We have a set of guidelines, organized by principles but not as happy paths.

We have updated scenarios.

We have defined research code & software.

A couple of definitions.

Research code: some code product that leads to some published outcome but is not a specific deliverable and usually not intended for reuse outside the project.

Research software: some code product that is intended for reuse outside the project (and is likely a specified deliverable).

We have different expectations for research code and for research software.

This impacts how we assess those products and how we talk about the sustainability of the products and the projects.

We also need to consider development cycles, funding cycles, and the lifetime of a project.

This is mostly a people problem so it’s about framing, about providing a coherent & actionable structure, and about doing so in ways that meet (most of) the needs of our diverse set of stakeholders.

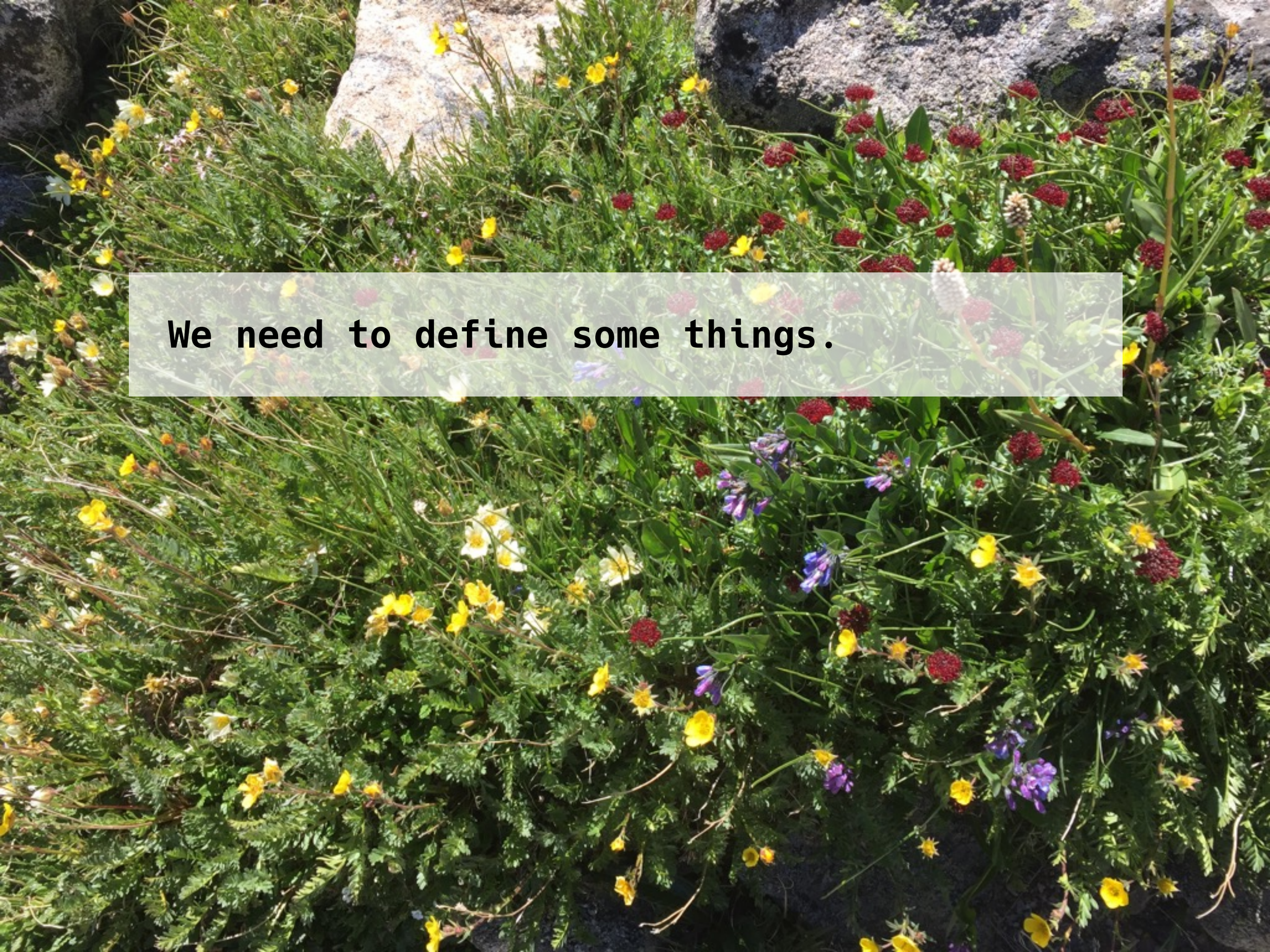
Our goals today:

Have a think about happy paths (at least for an ESIP Core set).

Have a harder think about our research situations (the scenarios defined here: https://titanpad.com/0z9RcazAJy), *maturities and sustainability.


Ready?

Progressions & Maturity



We need to define some things.

First.

The stages.

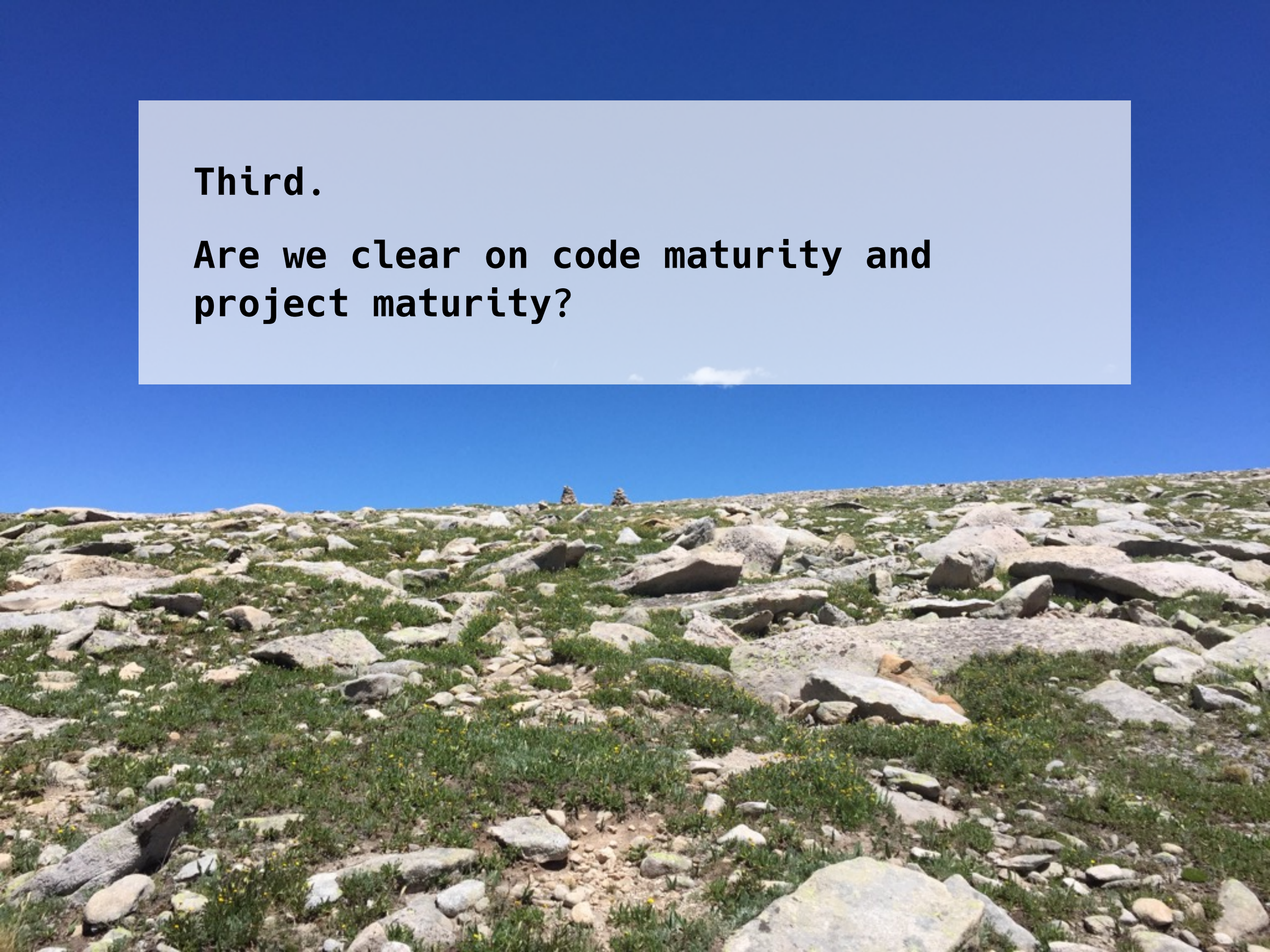
https://titanpad.com/F0TIRgokCq

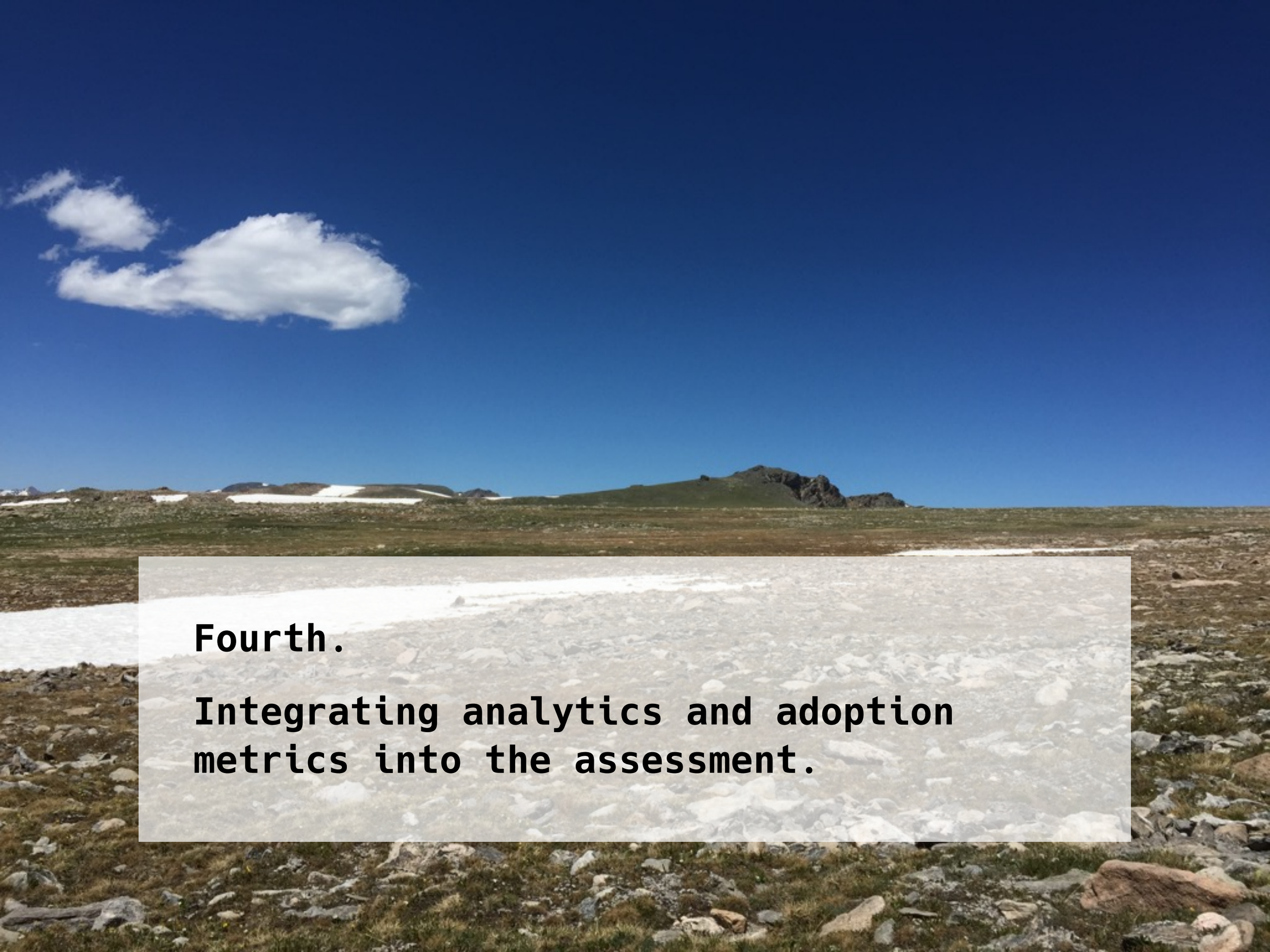
Second.

What principles (or sub-sections) are important at a given stage?

Third.

Are we clear on code maturity and project maturity?





Fourth.

Integrating analytics and adoption metrics into the assessment.

Related.

What are the service level metrics, assuming web services and applications we want to consider?

What can we not assess?

Sustainability

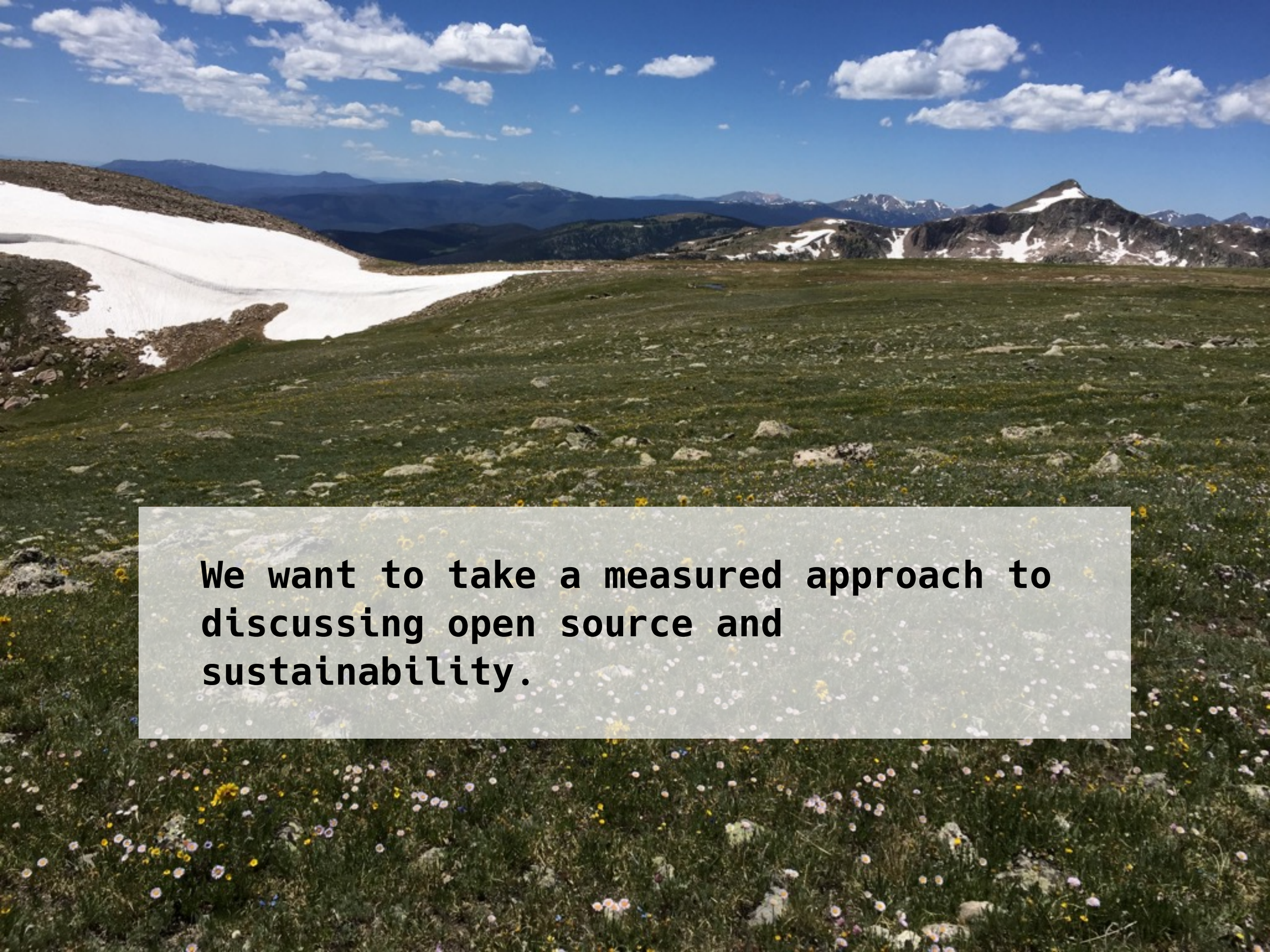
Sustainabilities?

Research code versus Research software

A note about Open Source…

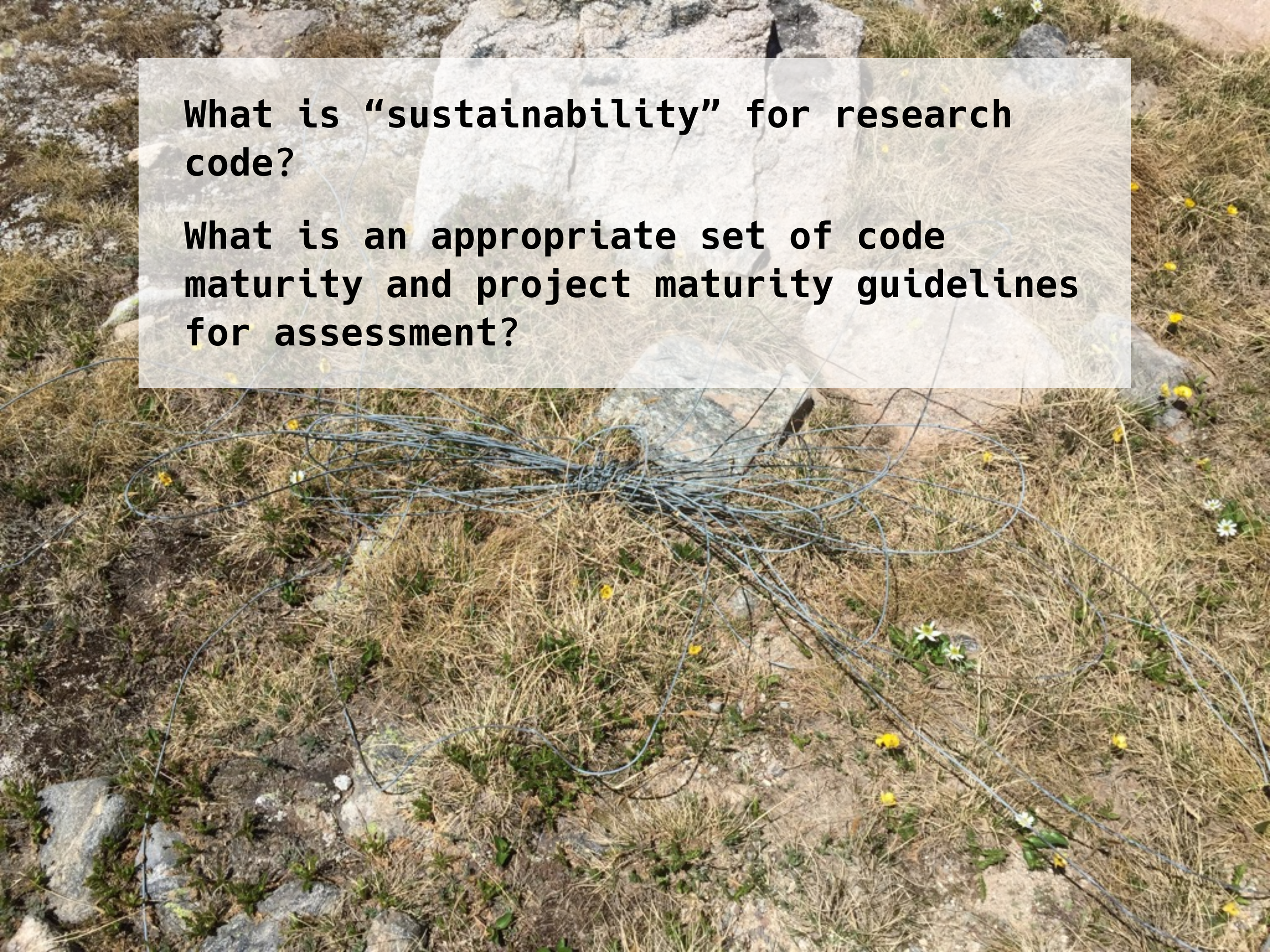
Roads and Bridges: The Unseen Labor Behind Our Digital Infrastructure

http://goo.gl/X0FvN2



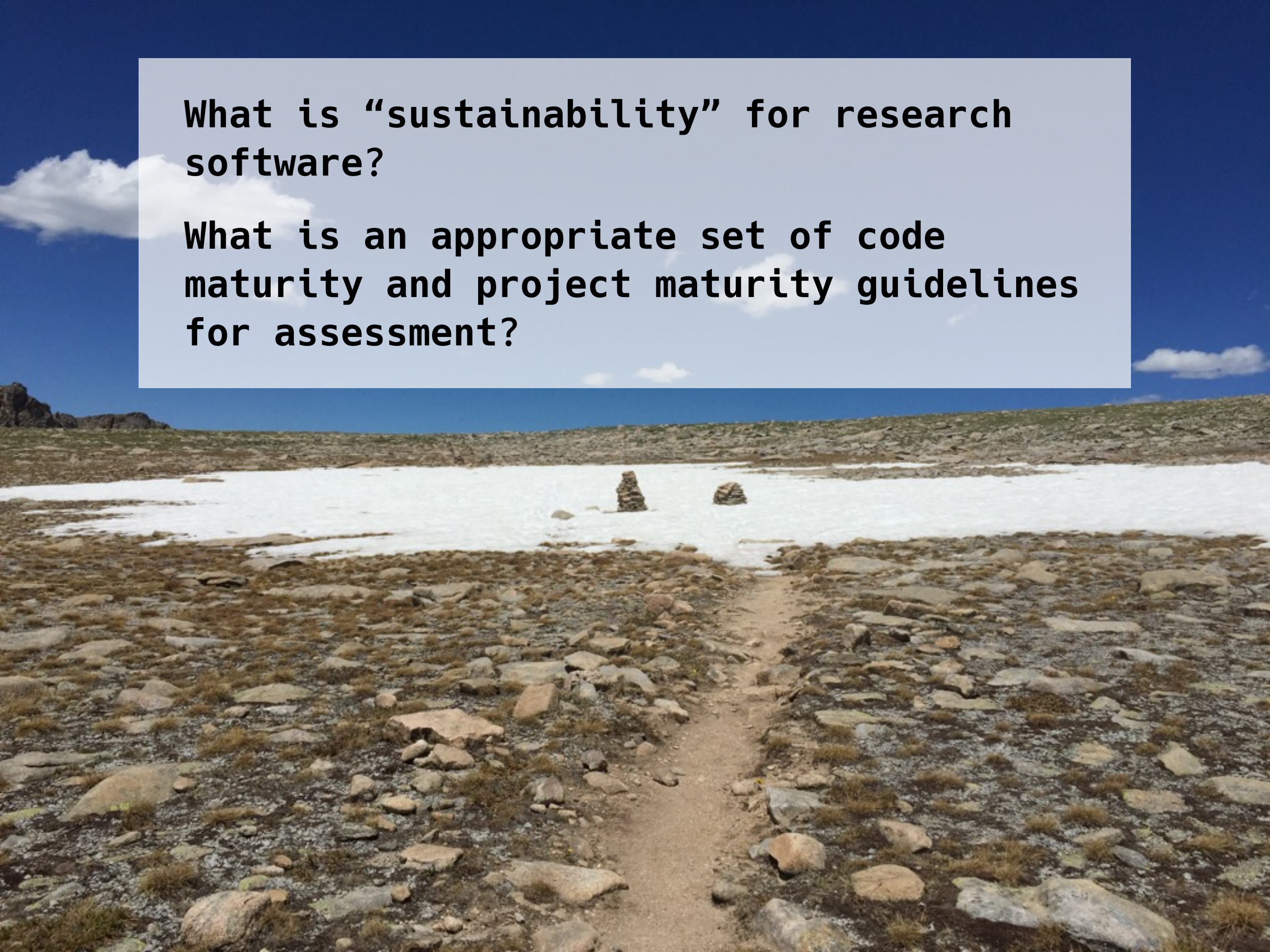
We want to take a measured approach to discussing open source and sustainability.

How do we frame the discussion as a continuum from Open Science to Open Source?



What is “sustainability” for research code?

What is an appropriate set of code maturity and project maturity guidelines for assessment?

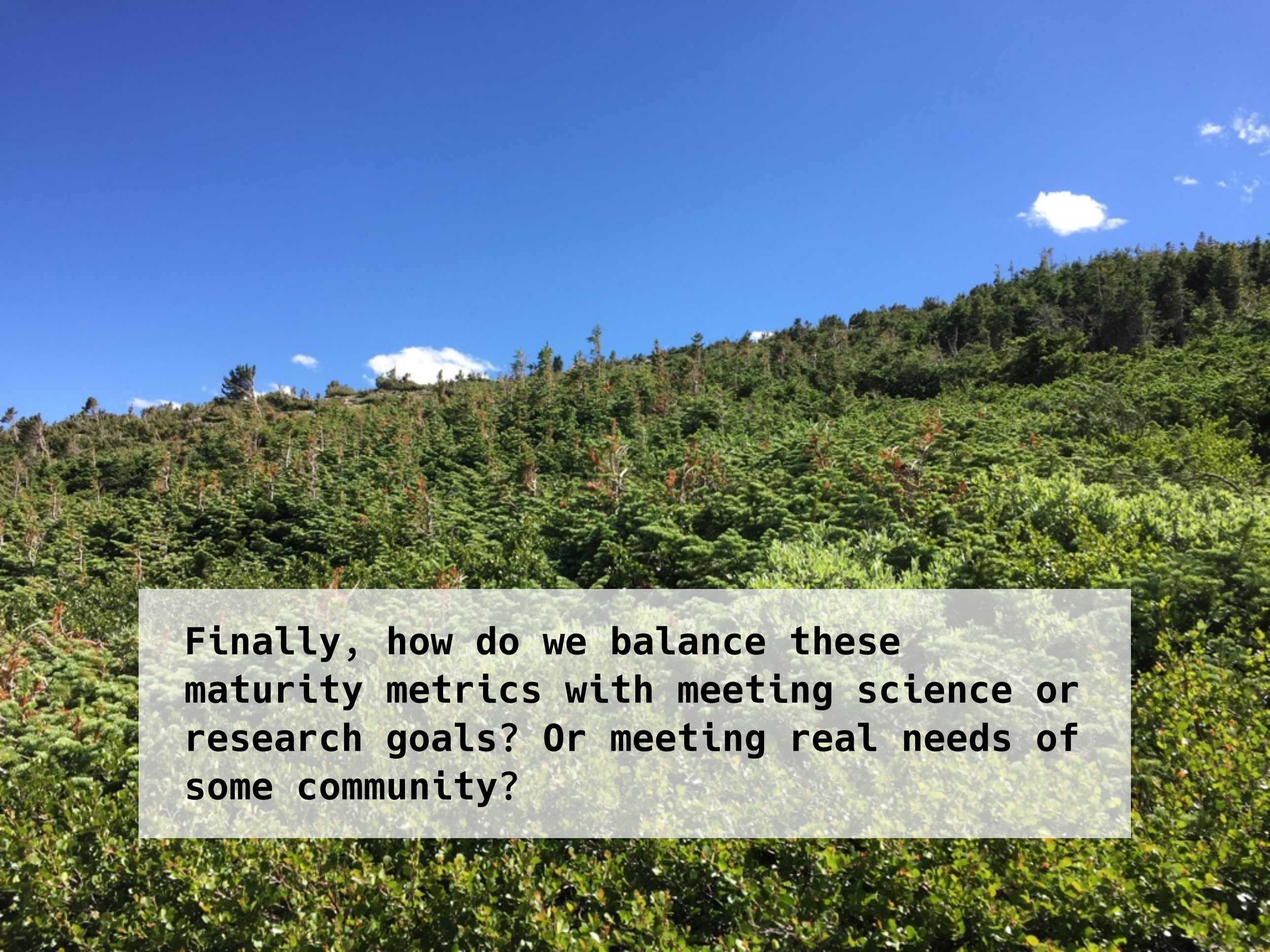


What is “sustainability” for research software?

What is an appropriate set of code maturity and project maturity guidelines for assessment?

From these progressions, is there anything stakeholder-specific that we need to address?

After this discussion, do we need to tease out project maturity from sustainability?



Finally, how do we balance these maturity metrics with meeting science or research goals? Or meeting real needs of some community?

Next (potential) steps.

Thank you.