Cytomining All-hands

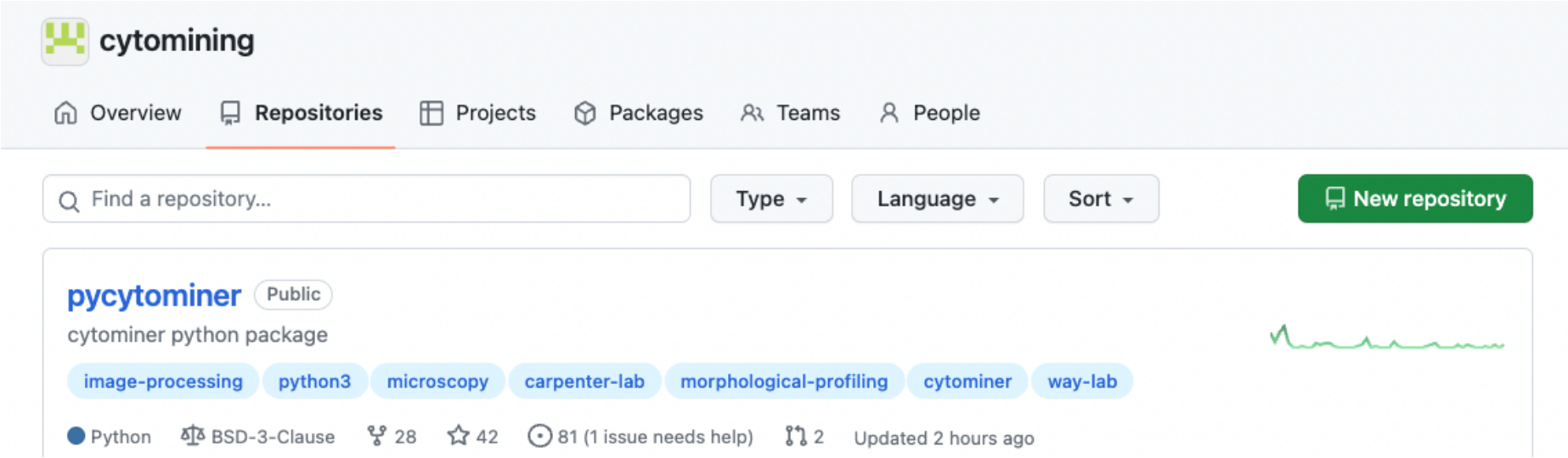
Envisoning future software efforts! 🚀

Gratitude!

Presentation Outline

1. 📚 Context
2. 📐 Measurements
3. 🔮 Strategies

Context - What is Cytomining?



Cytomining is a GitHub Organization which includes several projects the Way Lab develops and uses for research.

Link to Cytomining Github Org

Context - What projects are included?

- **Pycytominer**: “Pycytominer is a suite of common functions used to process high dimensional readouts from high-throughput cell experiments.”
- **DeepProfiler**: DeepProfiler is a set of tools to use deep learning for analyzing imaging data in high-throughput biological experiments.
- **Cytominer-eval**: Cytominer-eval contains functions to calculate quality metrics for perturbation profiling experiments.
- **CytoSnake** (wayscience repo): Reproducible pipelines for processing high-dimensional systems morphology data with snakemake
- **CytoTable**: CytoTable enables single-cell morphology data analysis by cleaning and transforming output data at scale.
- … and others!

Context - How big is Cytomining?

- Total number of repos: 25
- Total stars for cytomining repos: 212
- Total forks from cytomining repos: 161
- Total number of unique contributors: 36
- Total number of PyPI downloads (180 days): 18,672

Calculations reference

Measurements - Time

> *The key is not spending time but investing it.*

Steven R. Covey

Time

- 🪴 Software Garden Code Decay
- 🕑 Hofstadter’s Law
- 🔢 A rough calculation

Time - Code Decay

- 🪴 **Software Garden Code Decay**: your software garden is growing and decaying through time.
- Expect things to break as time goes on; the ecosystem changes fast!
- Because of this, we must be selective with how and what we spend our time on.

Time - Code Decay

An example: pycytominer#250

Time - Hofstadter's Law

- 🕑 **Hofstadter's Law**: "It always takes longer than you expect, even when you take into account Hofstadter's Law."
- Plan for delays; time taken will almost always be more than you expect.

Time - Hofstadter's Law - continued

- 🕑 Planning Fallacy: predictions about how much time it will take are often wrong and fall victim to optimism bias.

Time - Cytomining work stats

- Total open Cytomining repo issues: 230 (an estimate!)

Calculations reference

Time - A rough calculation

Let’s assume each open issue needs a minimum:

- **1 hour of development**
- **2 hours of pull-request review / responses**

Time - A rough calculation

That makes:

- 230 open issues * 3 hrs = **690 hours of work (minimum)**
- 690 hrs / 8 hrs = **86.25 full work days**
- 86.25 / 5 work days per week = **17.25 full work weeks**

😬

Measurements - Value

> *People think focus means saying yes to the thing you've got to focus on. But that's not what it means at all.* ***It means saying no to the hundred other good ideas that there are.*** *You have to pick carefully. I'm actually as proud of the things we haven't done as the things I have done. Innovation is saying no to 1,000 things.*

Steve Jobs

Value

- 💊 Vitamin, Painkiller, or Cure?
- ⚖️ Opportunity Cost
- 🗺️ Value Streams

Value - Vitamin, Painkiller, or Cure

- 💊 Will your software solution(s) be a vitamin, painkiller, or cure?
- Vitamins: enhance or strengthen existing things.
- Painkiller: remove barriers or roadbumps.
- Cure: completely resolve or predict resolutions.

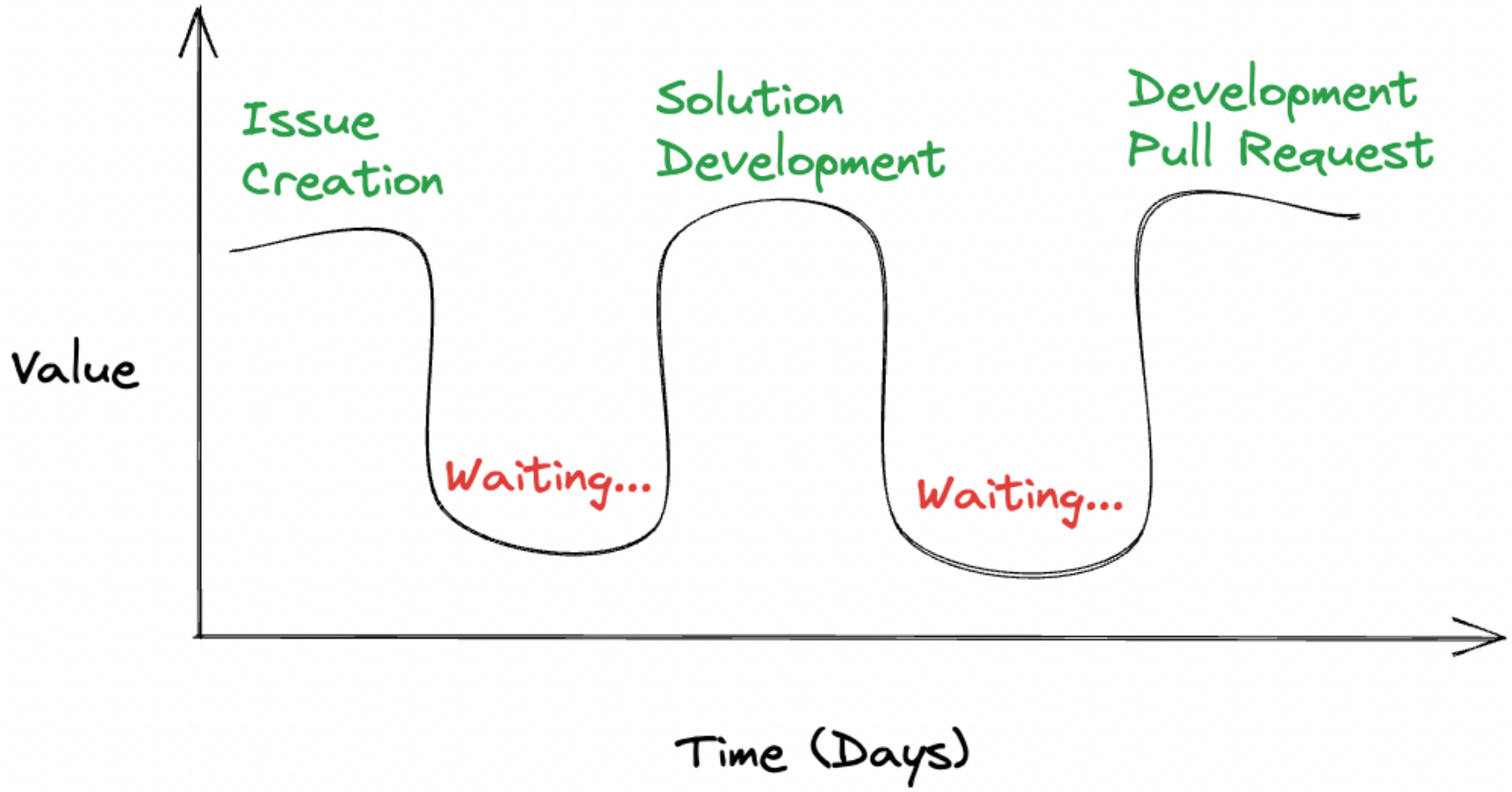
Value - Opportunity Cost

- ⚖️ **Opportunity Cost**: choosing one thing may mean giving up another.
- There are more software challenges than we have development time to accomplish.
- Consider what solutions you strive for by respecting what didn’t get attention.

Value - Value Streams

- 🗺️ **Value Streams**: the steps it takes from idea to value delivery to an audience.
- We often think about “waste” alongside value streams
- Value streams can get “blocked”, meaning things aren’t being completed.

Value - Value Streams



Value-added vs waiting time

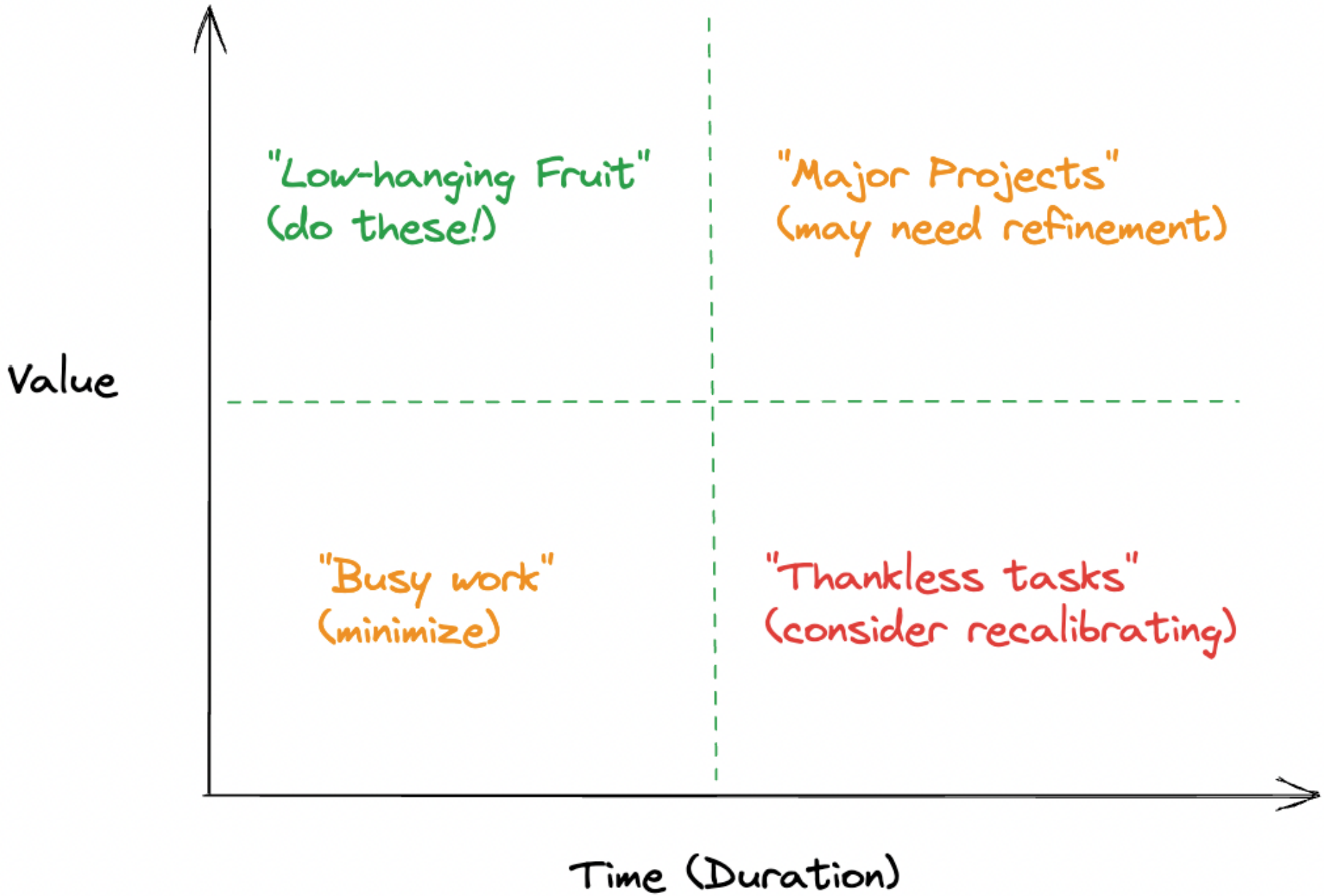
Strategies - Agile

> Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Agile Manifesto: Principles behind the Agile Manifesto

Strategies - Agile thinking

- Focus on modularization, breaking big problems into smaller ones
- Keep *kairos* in mind, or the opportune time to do things
- Use time and value to prefer high value, low time items


"Low-hanging Fruit"
(do these!)
"Major Projects"
(may need refinement)
"Busy work"
(minimize)
"Thankless tasks"
(consider recalibrating)
Value
Time (Duration)

Strategies - Backcasting

- High value, high time duration (“Major Projects”)
- Use Backcasting to “think with this end in mind”
- Backcasting: picturing the steps which take place before the end result

Strategies - Backtracing