

Open Project Management

from an “open” perspective

UNIT 5

Instructor: Dr. Bradly Alicea

<http://bradly-alicea.weebly.com>



Lecture 20

All content



Open Project Management

Welcome Back!



Project Lifecycle

Ubuntu releases

22.10 (Kinetic Kudu)

22.04 LTS (Jammy Jellyfish)

21.10 (Impish Indri)

21.04 (Hirsute Hippo)

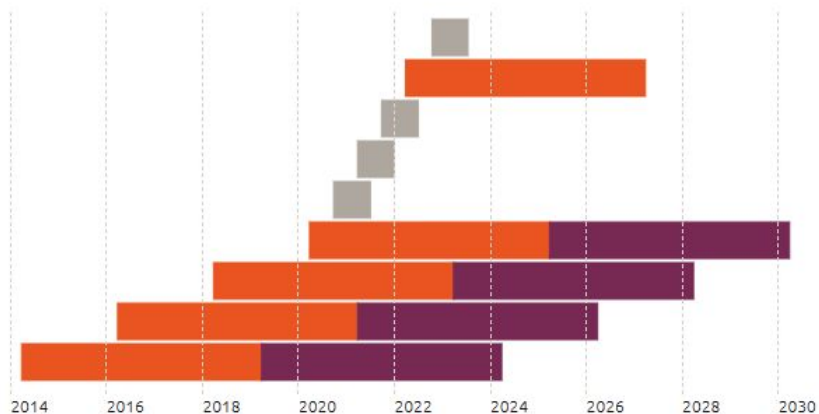
20.10 (Groovy Gorilla)

20.04 LTS (Focal Fossa)

18.04 LTS (Bionic Beaver)

16.04 LTS (Xenial Xerus)

14.04 LTS (Trusty Tahr)

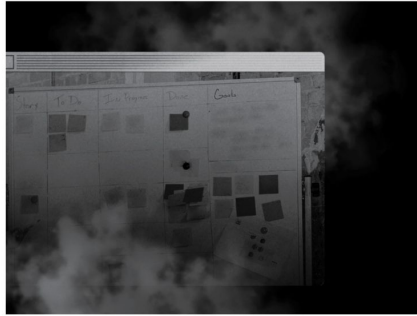


- Hardware and maintenance updates
- Interim release Standard Support
- Extended Security Maintenance (ESM)

Interim releases will introduce new capabilities from Canonical and upstream open source projects, they serve as a proving ground for these new capabilities. Many developers run interim releases because they provide newer compilers or access to newer kernels and newer libraries, and they are often used inside rapid devops processes like CI/CD pipelines where the lifespan of an artefact is likely to be less than the support period of the interim release. Interim releases receive full security maintenance for 'main' during their lifespan.

Ubuntu (Linux) Lifecycle

<https://ubuntu.com/about/release-cycle>



Agile and the Long Crisis of Software

Miriam Posner

What is Agile? And where does it come from?

Agile management processes and the rise of software development

<https://logicmag.io/clouds/agile-and-the-long-crisis-of-software/>

When Scrum was still in its infancy, a number of events contributed to its formation and its eventual, resulting form.

Jeff Sutherland, one of the agile development method's co-founders, formerly owned a company next to MIT. At the time, he would pluck young minds from MIT's Media Lab just as they were graduating. Once outside the lab, they brought with them a mantra that would go on to influence Scrum deeply. They called it "Demo or Die."

And as a result of Demo or Die, Scrum went on to encapsulate the idea that it's imperative, as part of a best practices process, to include a demonstration phase prior to a roll-out. It was simply a necessary facet.

Origins of "Demo or Die" and Retrospection via Sprint Review

<https://openviewpartners.com/blog/the-origin-of-demo-or-die-and-retrospection-via-sprint-review/>

My team, after that meeting, they would all put their heads on the table, and they would say: “We don’t know if we can do another one of these demos.”

“You have a choice. You can be just another software development team or you can be a great team, and to be great you need this feedback.”

Qualitative Assessment of Demo or Die

My team, after that meeting, they would all put their heads on the table, and they would say: “We don’t know if we can do another one of these demos.”

“You have a choice. You can be just another software development team or you can be a great team, and to be great you need this feedback.”

“Okay. We’ll do one more demo.” So you can see how critical this would be to build a really cool product, having that dynamic feedback from people who really understand what needs to be built.

My team, after that meeting, they would all put their heads on the table, and they would say: “We don’t know if we can do another one of these demos.”

“You have
a great tea

Demo or Die session post-mortem

ou can be

“Okay. We
really cool product, having that dynamic feedback from people who really understand what needs to be built.

How can we do better in the next cycle? That review has come to be called a retrospective, where the team goes through, “Okay. What have we done? What did we like? What didn’t we like? What are the process improvements that we can make?” So the retrospective follows right after what we have come now to call the sprint review or the demo of the product.



THE FUTURE OF THUNDERBIRD

MODERNIZING AN ANCIENT APPLICATION

How we got to this point?

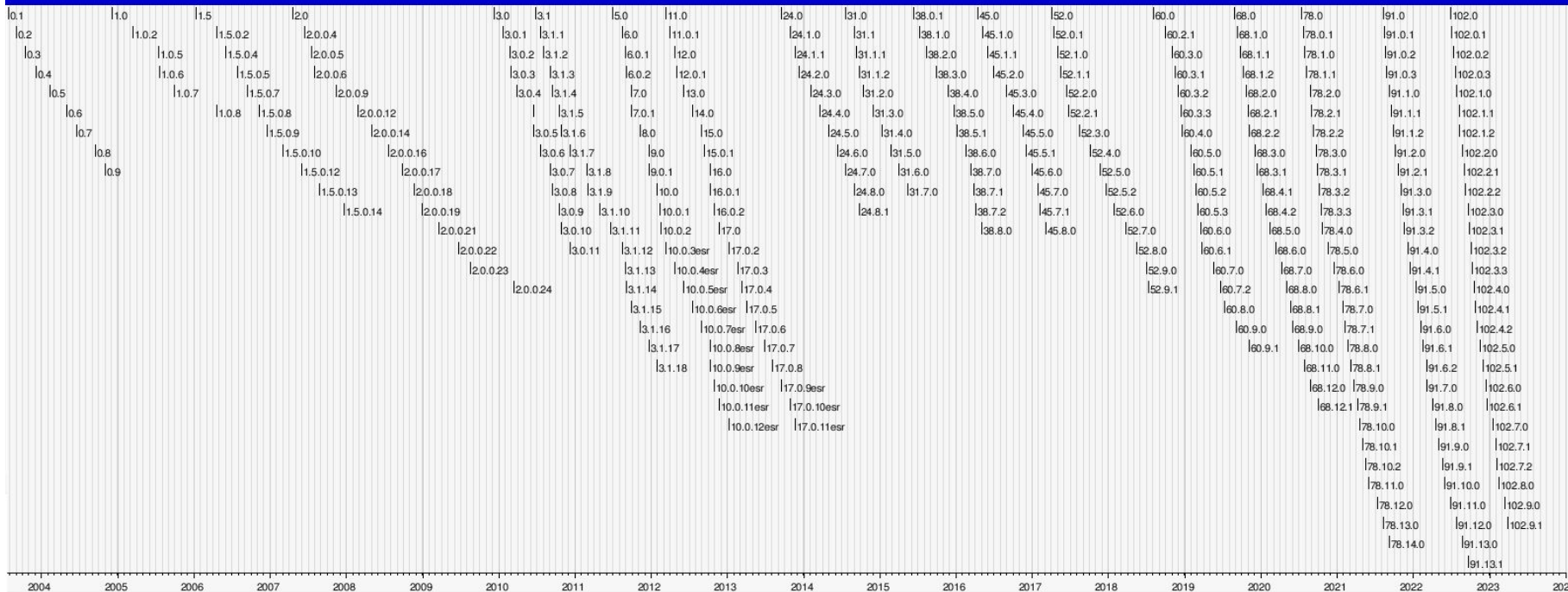
What's our plan?

What needs to happen?

The Future of Thunderbird - Modernizing an Ancient Application

Alessandro Castellani

<https://www.youtube.com/watch?v=EoLb6aHakno>



https://en.wikipedia.org/wiki/History_of_Mozilla_Thunderbird

The Digital Economy Runs on Open Source. Here's How to Protect It

Harvard Business Review <https://hbr.org/2021/09/the-digital-economy-runs-on-open-source-heres-how-to-protect-it>

Security and Sustainability issues with long-term open-source development. Companies should proactively support FOSS to ensure resources will be there.

The Digital Economy Runs on Open Source. Here's How to Protect It

Harvard Business Review <https://hbr.org/2021/09/the-digital-economy-runs-on-open-source-heres-how-to-protect-it>

Security and Sustainability issues with long-term open-source development. Companies should proactively support FOSS to ensure resources will be there.

What happens when large corporations acquire an open-source project?

- repackage and sell product (change terms of access).
- conflict with previous contributors' work, previous licenses.

The Digital Economy Runs on Open Source. Here's How to Protect It

Harvard Business Review <https://hbr.org/2021/09/the-digital-economy-runs-on-open-source-heres-how-to-protect-it>

Security and Sustainability issues with long-term open-source development. Companies should proactively support FOSS to ensure resources will be there.

What happens when large corporations acquire an open-source project?

- repackage and sell product (change terms of access).
- conflict with previous contributors' work, previous licenses.

Moving from small group projects to formal ones?

- software archives hosted on individual accounts (not a permanent solution).
- outdated versions still in use, underlie other major projects.

Open-source Maturity Model

General structure [\[edit \]](#)

OMM is organized in levels, each level is building on and including the trustworthy elements (TWE) at the lower level. The trustworthy elements included in OMM were collected or inspired by two sources:

1. FLOSS-TWEs gathered from an extensive survey conducted on FLOSS developers, FLOSS users, and FLOSS integrators^[9]
2. [CMMI Process Areas](#)

Basic level [\[edit \]](#)

The **basic level** that can be easily reached by adopting a few necessary practices in the FLOSS development process:

- **PDOC** – Product Documentation
- **STD** – Use of Established and Widespread Standards
- **QTP** – Quality of Test Plan
- **LCS** – Licenses
- **ENV** – Technical Environment
- **DFCT** – Number of Commits and Bug Reports
- **MST** – Maintainability and Stability
- **CM** – Configuration Management
- **PP1** – Project Planning Part 1
- **REQM** – Requirements Management
- **RDMP1** – Availability and Use of a (product) roadmap

Intermediate level [\[edit \]](#)

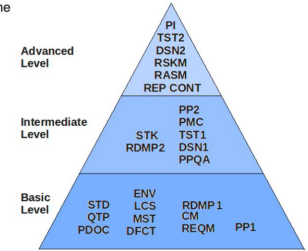
The **intermediate level** is the second level in OMM and can be achieved by fulfilling all trustworthy elements from the basic level and required trustworthy elements from the intermediate level.

- **RDMP2** – Availability and Use of a (product) roadmap
- **STK** – Relationship between Stakeholders
- **PP2** – Project Planning Part 2
- **PMC** – Project Monitoring and Control
- **TST1** – Test Part 1
- **DSN1** – Design Part 1
- **PPQA** – Process and Product Quality Assurance

Advanced level [\[edit \]](#)

The **advanced level** is the highest level that FLOSS projects can achieve by fulfilling all trustworthy elements from basic and intermediate levels and required trustworthy elements from the advanced level.

- **PI** – Product Integration
- **RSKM** – Risk Management
- **TST2** – Test Part 2
- **DSN2** – Design 2
- **RASM** – Results of third party assessment
- **REP** – Reputation
- **CONT** – Contribution to FLOSS Product from SW Companies

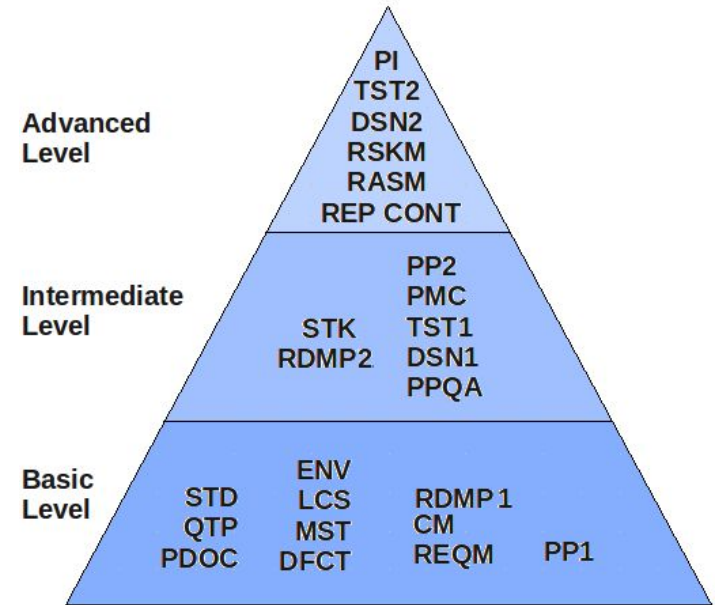


Open-source Maturity Model

Basic level [\[edit \]](#)

The **basic level** that can be easily reached by adopting a few necessary practices in the FLOSS development process:

- **PDOC** – Product Documentation
- **STD** – Use of Established and Widespread Standards
- **QTP** – Quality of Test Plan
- **LCS** – Licenses
- **ENV** – Technical Environment
- **DFCT** – Number of Commits and Bug Reports
- **MST** – Maintainability and Stability
- **CM** – Configuration Management
- **PP1** – Project Planning Part 1
- **REQM** – Requirements Management
- **RDMP1** – Availability and Use of a (product) roadmap

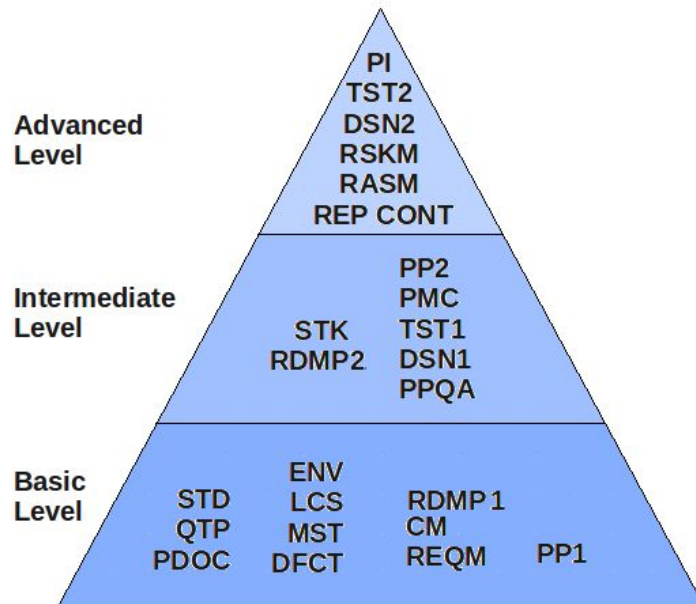


Open-source Maturity Model (con't)

Intermediate level [\[edit \]](#)

The **intermediate level** is the second level in OMM and can be achieved by fulfilling all trustworthy elements from the basic level and required trustworthy elements from the intermediate level.

- **RDMP2** – Availability and Use of a (product) roadmap
- **STK** – Relationship between Stakeholders
- **PP2** – Project Planning Part 2
- **PMC** – Project Monitoring and Control
- **TST1** – Test Part 1
- **DSN1** – Design Part 1
- **PPQA** – Process and Product Quality Assurance

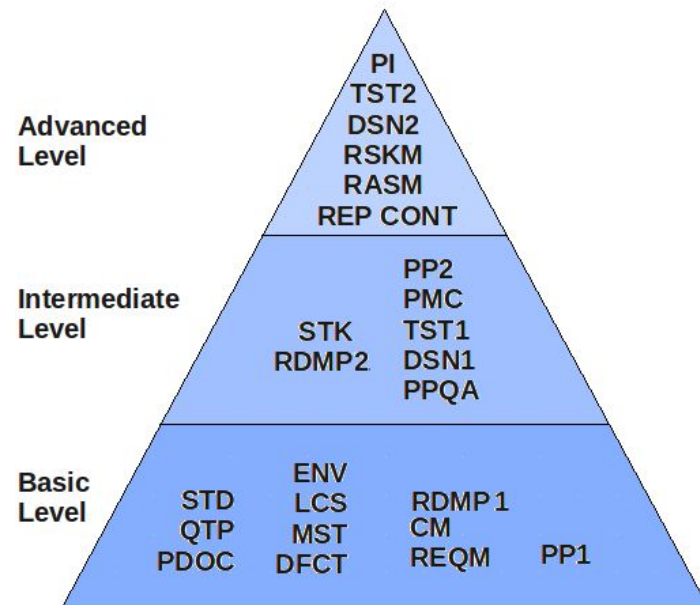


Open-source Maturity Model (con't)

Advanced level [\[edit \]](#)

The **advanced level** is the highest level that FLOSS projects can achieve by fulfilling all trustworthy elements from basic and intermediate levels and required trustworthy elements from the advanced level.

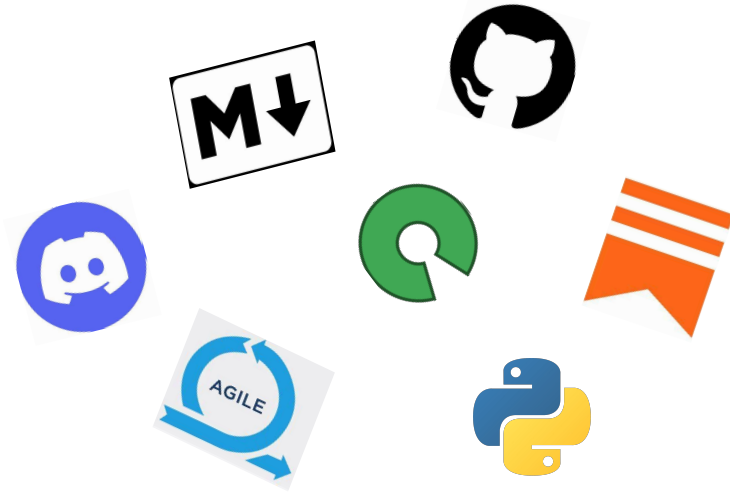
- **PI** – Product Integration
- **RSKM** – Risk Management
- **TST2** – Test Part 2
- **DSN2** – Design 2
- **RASM** – Results of third party assessment
- **REP** – Reputation
- **CONT** – Contribution to FLOSS Product from SW Companies



User Path Analysis

A more quantitative way to understand pathways is to create a user path analysis.

A user path analysis involves evaluating the different paths through your community's various resources most frequented by contributors and participants.



User Path Analysis (con't)

A more quantitative way to understand pathways is to create a user path analysis.

A user path analysis involves evaluating the different paths through your community's various resources most frequented by contributors and participants.

For example, a common pathway might be from Onboarding Guide to Wiki to Guidance Tree to Github Repository.

User Path Analysis (con't)

A more quantitative way to understand pathways is to create a user path analysis.

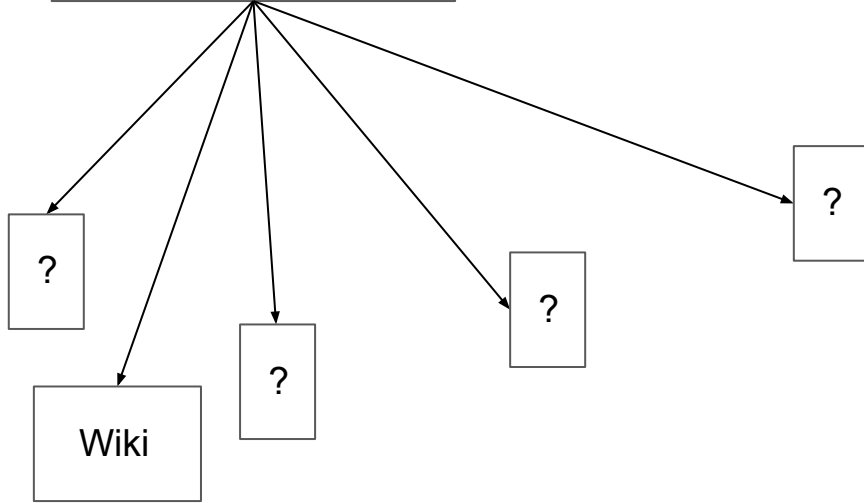
A user path analysis involves evaluating the different paths through your community's various resources most frequented by contributors and participants.

For example, a common pathway might be from Onboarding Guide to Wiki to Guidance Tree to Github Repository.

You might use this information to optimize this pathway and expect a large audience at every step.

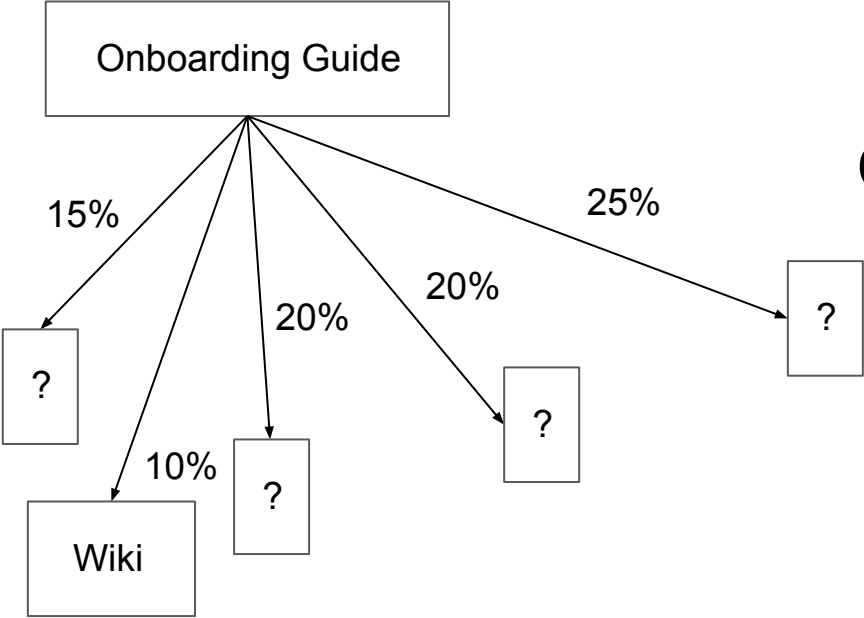
Alternatively, this pathway might be the most common, but is only taken by about 10% of your contributors. In such cases, you might look to cultivate alternate pathways.

Onboarding Guide

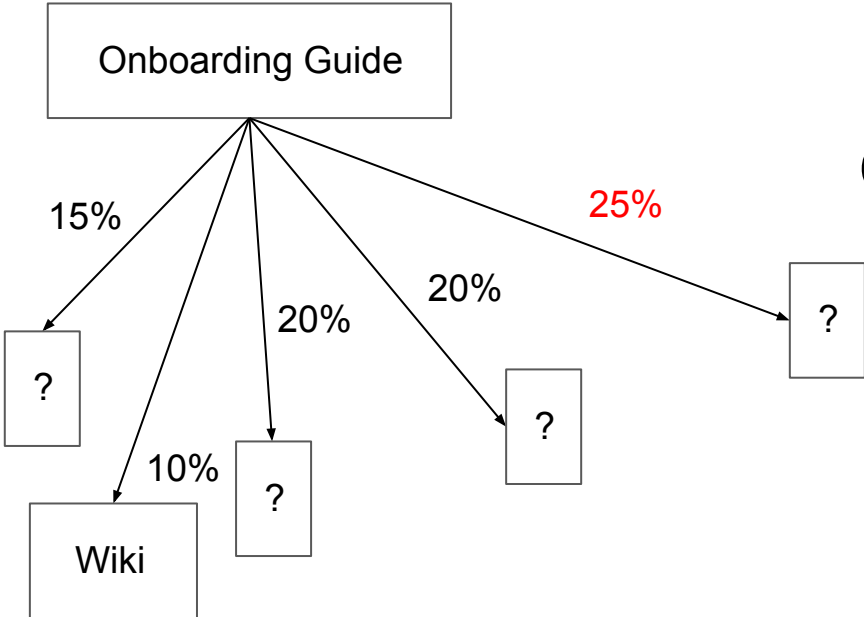


User Paths (from Onboarding to Success)

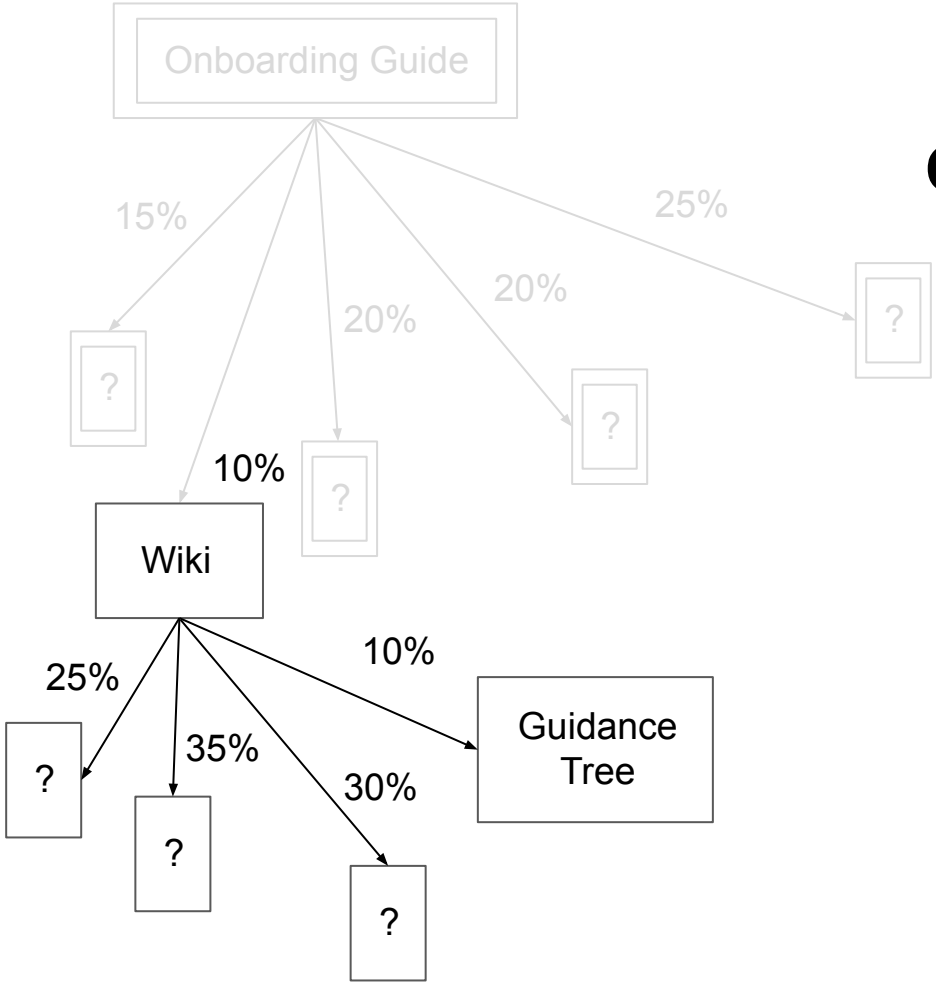
User Paths (from Onboarding to Success)



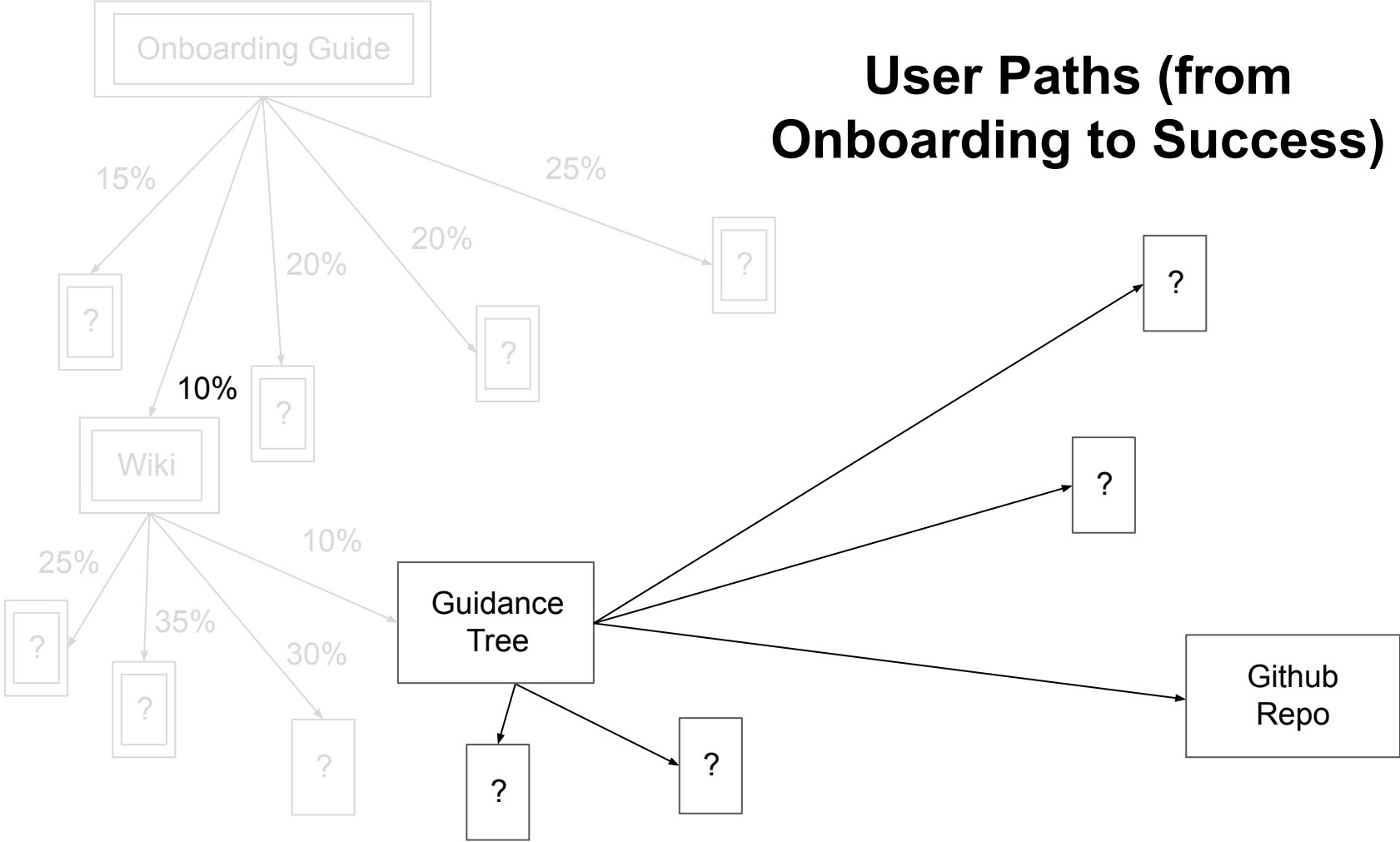
User Paths (from Onboarding to Success)



User Paths (from Onboarding to Success)



User Paths (from Onboarding to Success)



User Path Analysis (con't)

Path analysis can also help prioritize which community resources need to be maintained on a high-frequency basis, and which resources can be maintained less frequently (or replaced by different resources).

Is path analysis a good use of your time?

<https://www.kaushik.net/avinash/path-analysis-a-good-use-of-time/>

Can discover the sequence of resources used, but not why a particular path is taken.

- some paths are important for specific subgroups of contributors (personas).
- Are your resources well connected? Can contributors move from onboarding to discussions to code repository (Github)?

What are the hurdles to contribution?

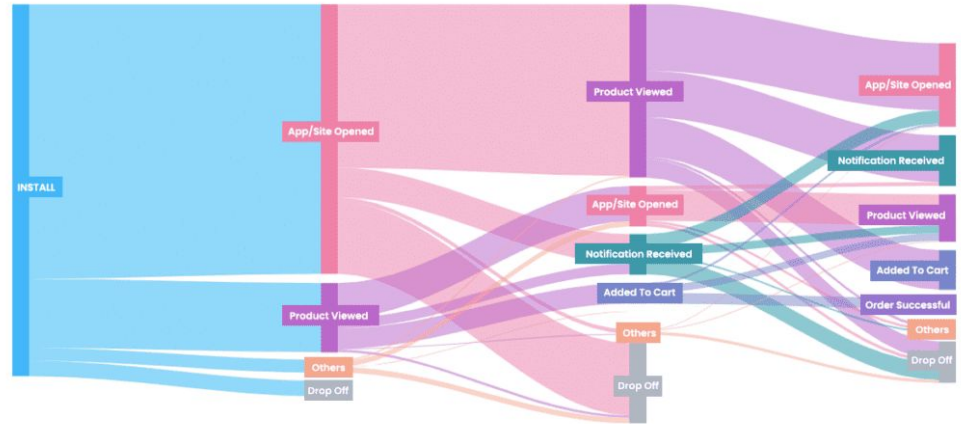
- do you need additional resources, or to consolidate the resources currently in existence?

6 issues that user path analysis can uncover

<https://www.moengage.com/blog/issues-user-path-analysis-can-uncover/>

Allows you to identify

- where people abandon onboarding process.
- Build a smooth path from onboarding to successful contribution.
- measure churn and diagnose contributor loss.



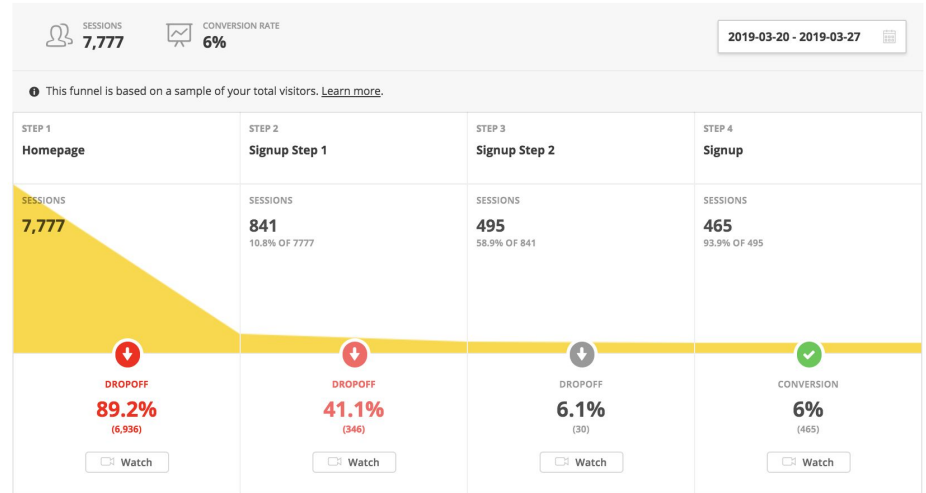
Additional Resources on Path Analysis

User Path Analysis (Google Analytics)

<https://support.google.com/analytics/answer/9317498?hl=en>

Path Analysis in Computing

[https://en.wikipedia.org/wiki/Path_analysis_\(computing\)](https://en.wikipedia.org/wiki/Path_analysis_(computing))



Funnel Analysis (example)

https://en.wikipedia.org/wiki/Funnel_analysis#/media/File:Funnel-analysis-example.png