Choice Matters: Contrasting Package Manager User Experience

Syful Islam

Raula Gaikovina Kula

Bodin Chinthanet

Christoph Treude

Takashi Ishio

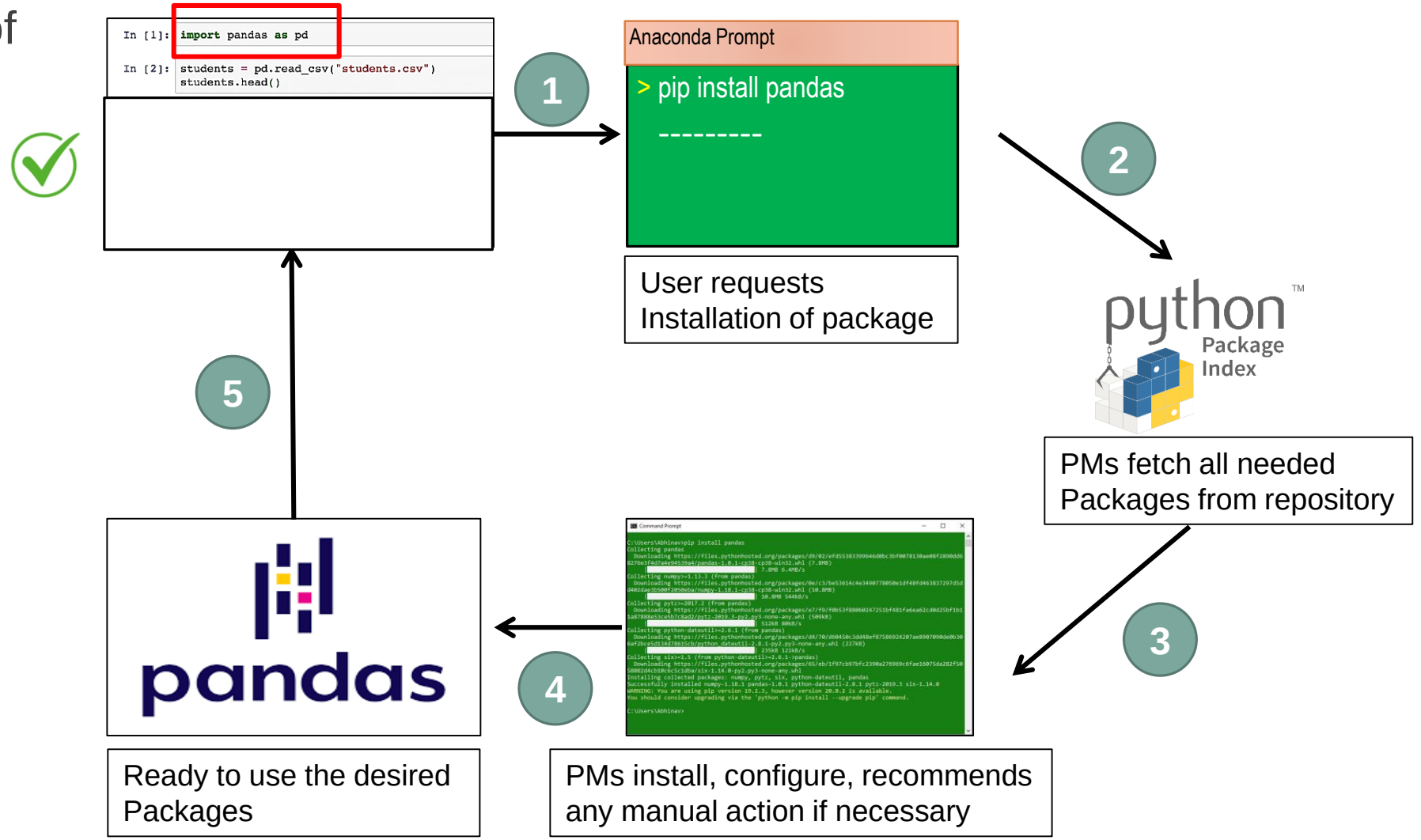
Kenichi Matsumoto

NAIST

THE UNIVERSITY of ADELAIDE

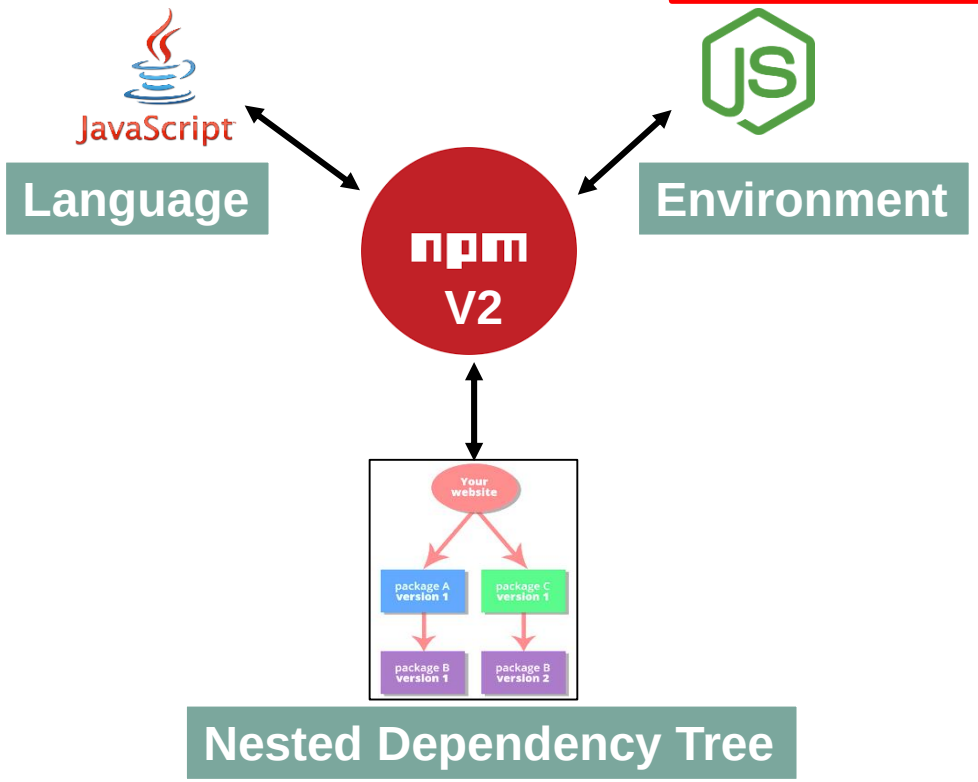
Package Managers are Crucial to Most Technology Stacks

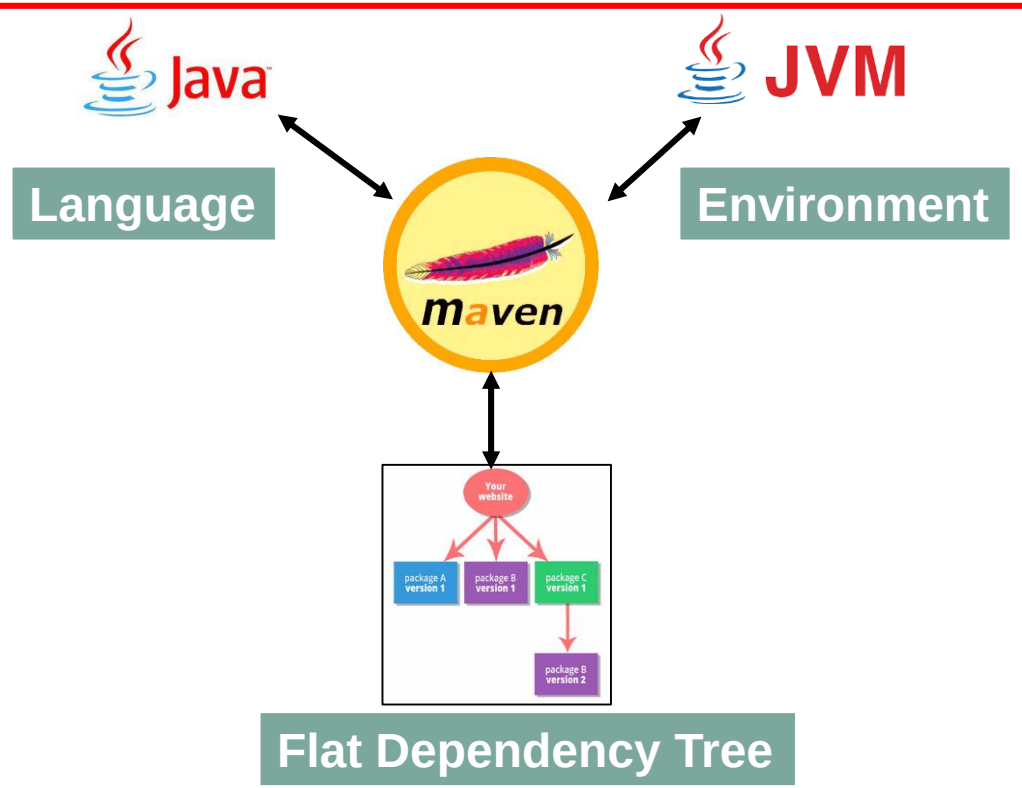
- Automates the process of
  - Installing
  - Upgrading
  - Configuring and
  - Removing of packages



Diversity of Technology Stacks has Led to Variety of Package Managers

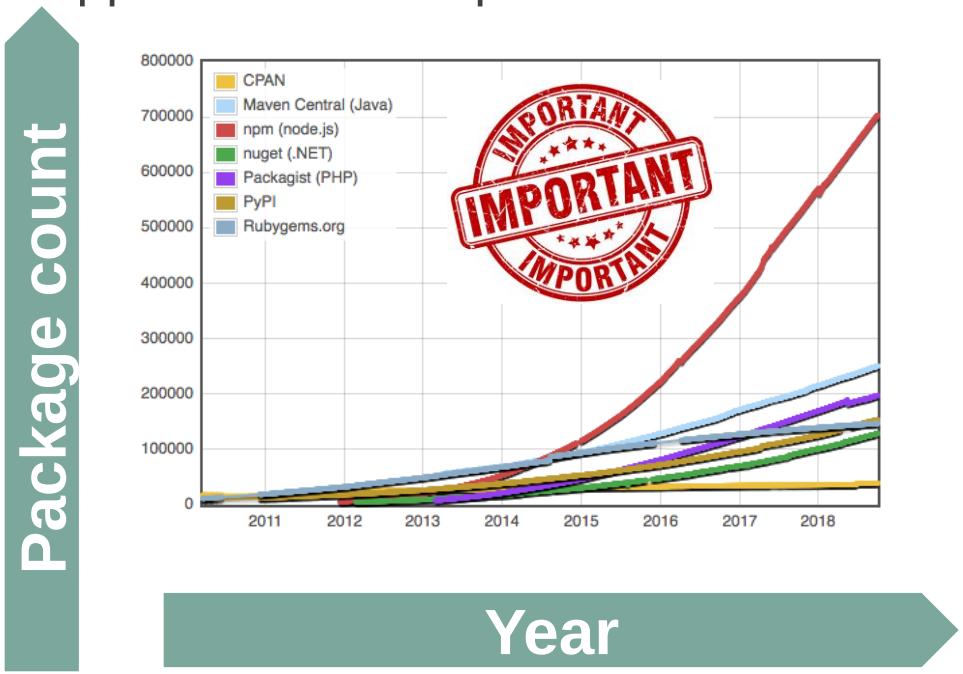
- Package Managers varies
  - Language
  - Environment
  - Dependency tree





Package Managers are Automated Solution for Applications that Rely on Third-party Packages

- Package Managers act as a broker of packages
  - Web building &
  - Mobile application development



Serve over 5 million open source packages

Ensuring the integrity and authenticity of the package

Grouping packages by function to reduce user confusion

Earlier in 2020, GitHub acquired the npm, which is the largest PM serving over 1.3 million packages to roughly 12 million developers, and constantly growing each day

[2]. https://blog.npmjs.org/post/180868064080/this-year-in-javascript-2018-in-review-and-npms

Related Research and Gap

- Studies on package dependency update
  - Bogart et al. investigates the reasons why developers do not update dependencies [1]
    - Ecosystem community values such as policies, supporting infrastructure, and accepted trade-offs to negotiate dependency changes
  - Kula et al. found that 69% of the developers are unaware of the need to update dependency and perceived extra workload [2].
  - Dietrich et al. reported that developers are facing challenge on which version of package to depend [3].
  - Developers are struggling to migrate their dependent packages
- Research Gap
  - Still it is unclear whether choice of Package manager is a matter to vary users experience

1. Bogart, Christopher, et al. "How to break an API: cost negotiation and community values in three software ecosystems." Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering. 2016.

2. Kula, Raula Gaikovina, et al. "Do developers update their library dependencies?." Empirical Software Engineering 23.1 (2018): 384-417.

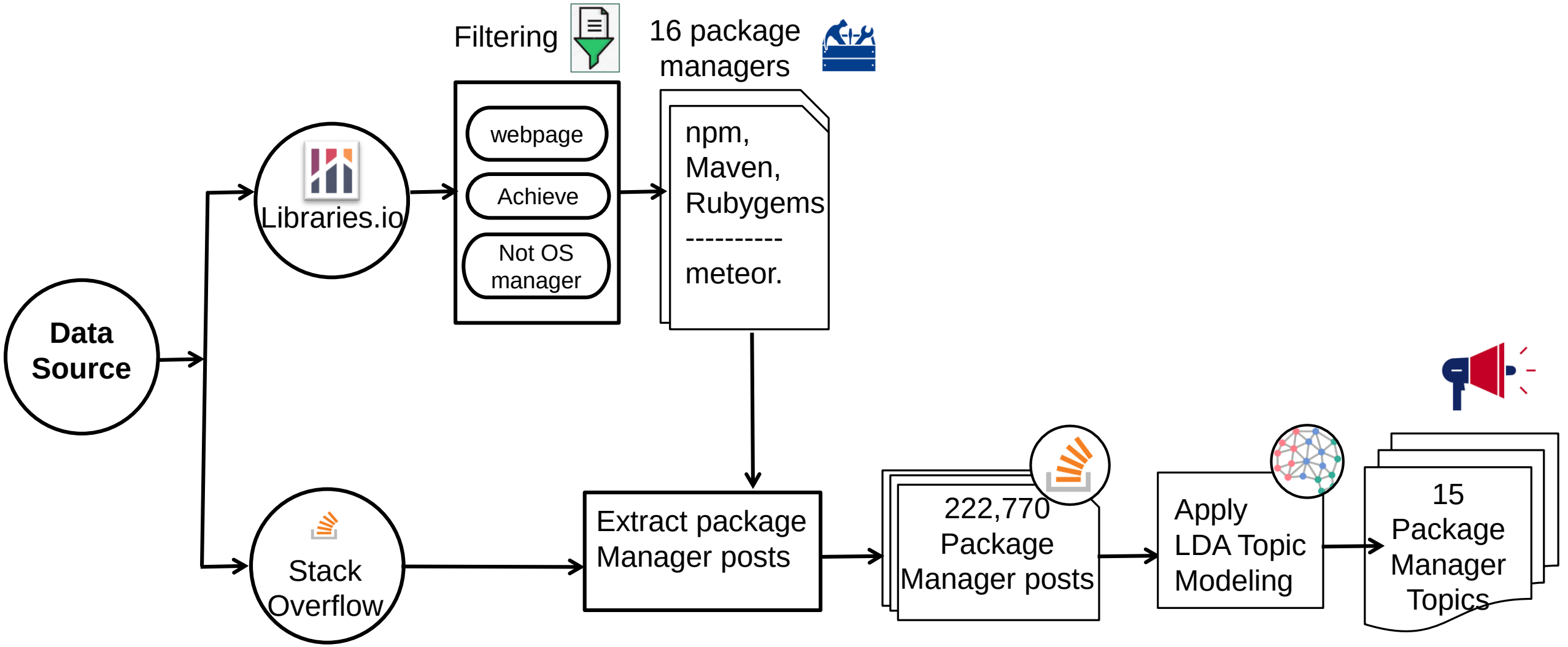
3. Dietrich, Jens, et al. "Dependency versioning in the wild." 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR). IEEE, 2019.

Objective and Study Design

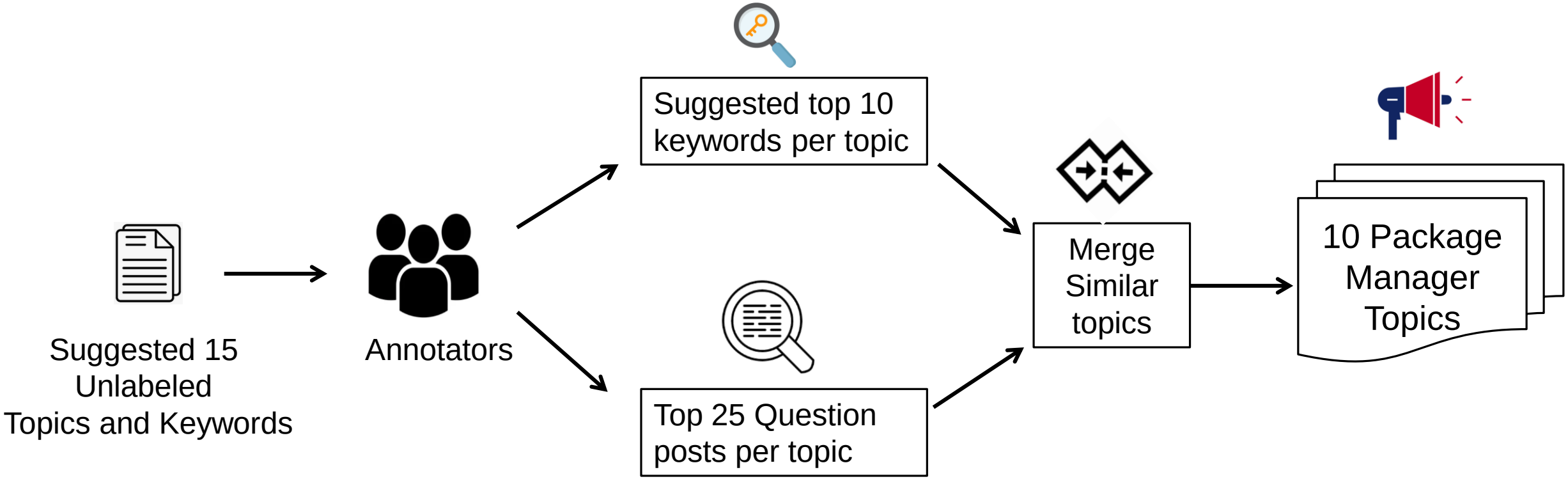
- Objective
  - Contrasting package managers related Stack Overflow Posts to investigate how the choice of package manager tool impact the users experience.

- Study Design

Collecting Package Manager Posts and Topic Modeling



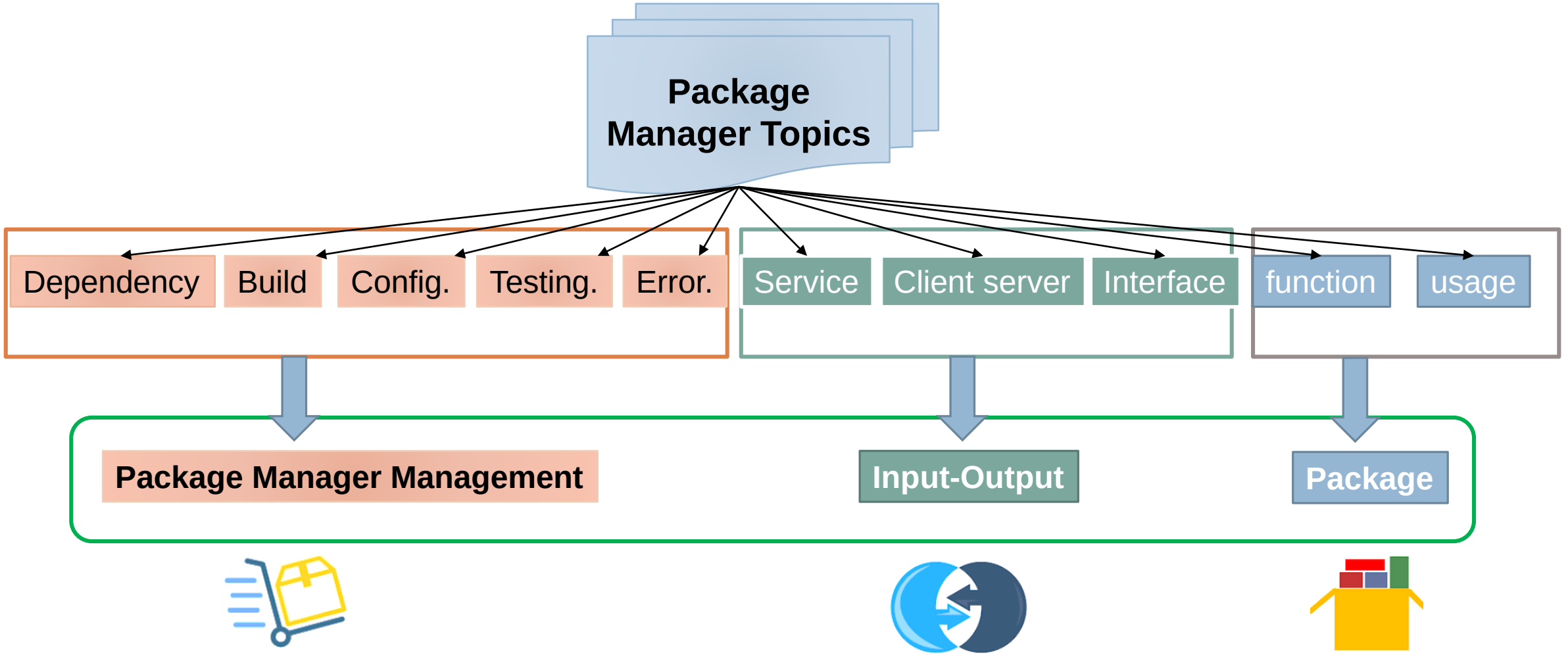
Label Package Manager Topics



Package Manager Topics [1/2]

Topics	Top 10 Keywords
Dependency	Dependency, version, specific, release, artifact, resolve, late, update, change, repository
Build	Build, project, create, generate, make, jar, war, compile, failure, resource
Configuration	Package, add, reference, install, set, environment, variable, import, module, library
Testing	Run, test, command, fail, execute, integration, unit, report, clean, surefire
Error	Error, throw, give, exception, load, unable, fix, issue, work, problem
Service	Application, deploy, wen, spring, app, tomcat, deployment, service, engine, boot
Client-server	Server, client, user, request, http, connection, access, response, message, proxy
Interface	File, time, read, write, image, output, channel, log, process, multiple
Package function	Type, function, struct, string, variable, interface, slice, method, argument, pass
Package usage	Template, react, collection, event, render, helper, database, component, field, document

Package Manager Topics [2/2]



Characterizing Package Manager users experience

Package Manager Users Experience

- **Topic**
- **Feature**
- **Response**

Topic mapping

Post	PM	Topic	Theme
1	npm	Error	PM
2	maven	Testing	PM

Feature mapping

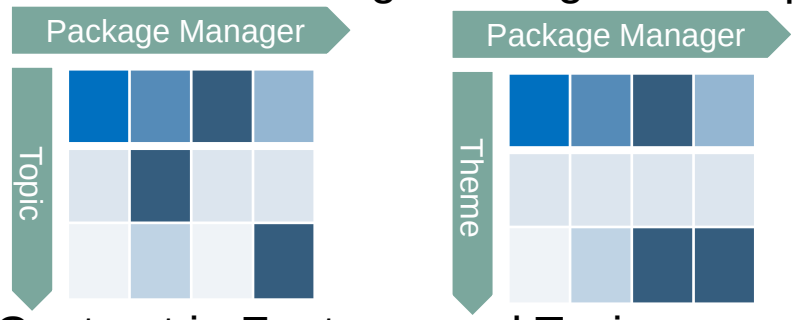
Post	Topic	Env.	Tree
1	Error	Node.js	Nested
2	Testing	JVM	Flat

Response mapping

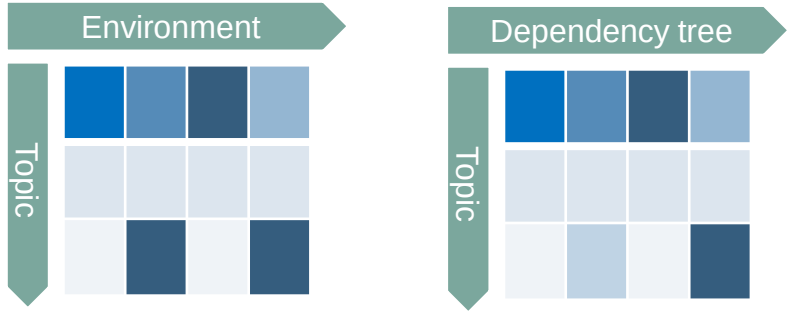
Post	Topic	# Score	# views	# Fav.	#Ans	#Acc ans
1	Error	20	1570	25	10	0
2	Testing	45	3238	300	5	1

Yang et al. (2016)

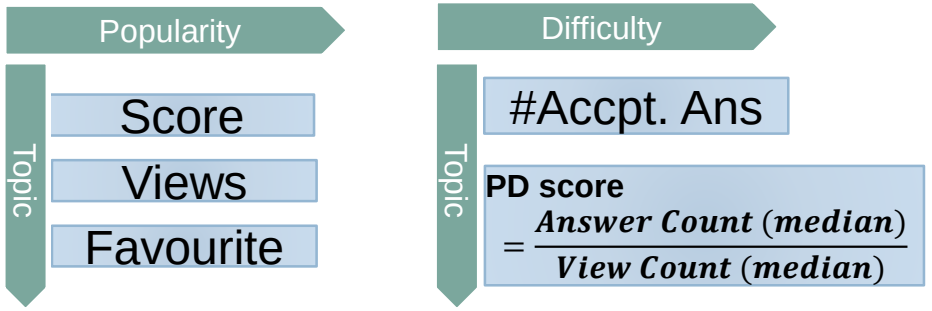
Contrast in Package Manager and Topics



Contrast in Features and Topics

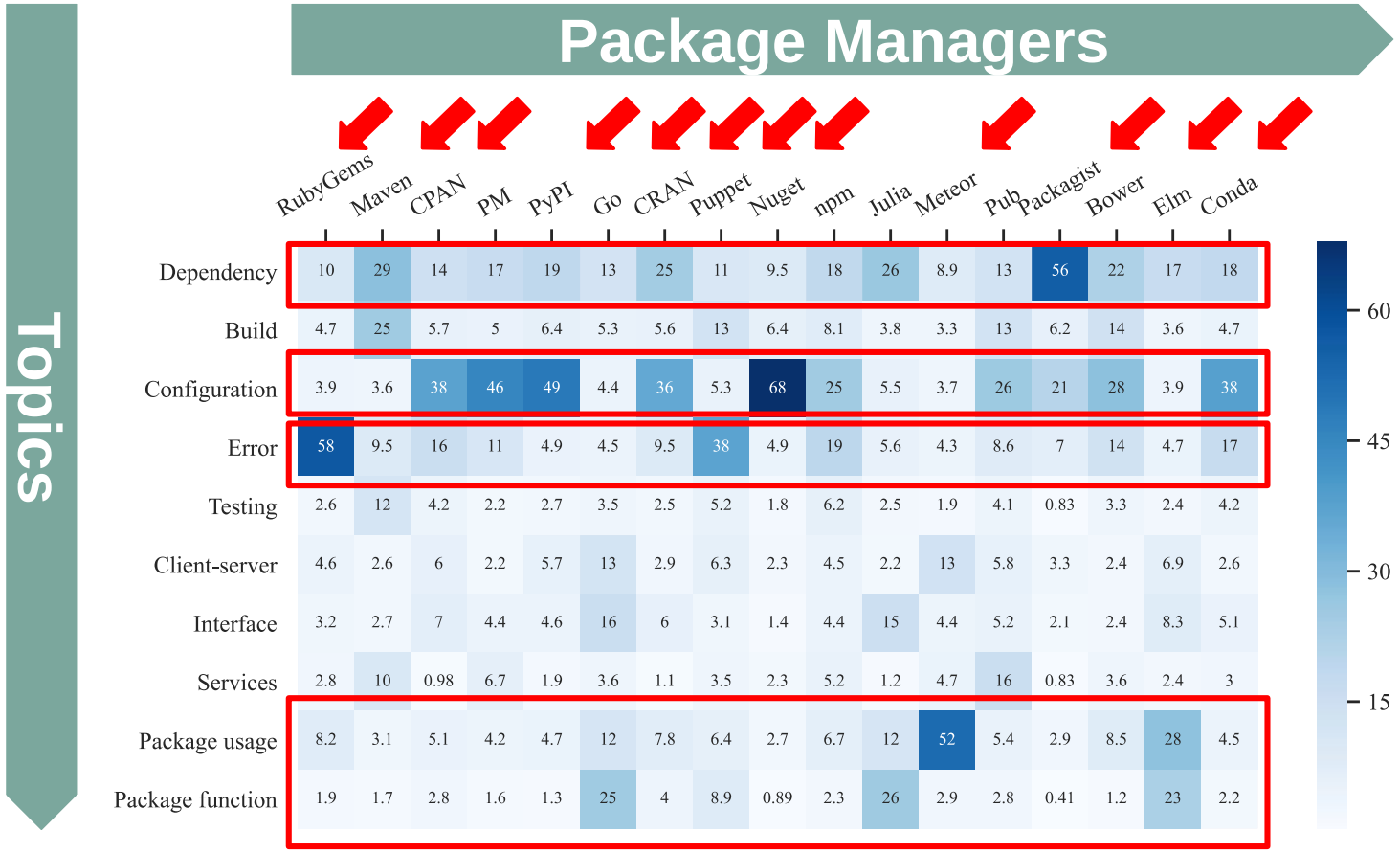


Contrast in Responses



1. Yang, Xin-Li, et al. "What security questions do developers ask? a large-scale study of stack overflow posts." *Journal of Computer Science and Technology* 31.5 (2016): 910-924.

Contrasts in Topics:
Topics and Package Managers [1/2]



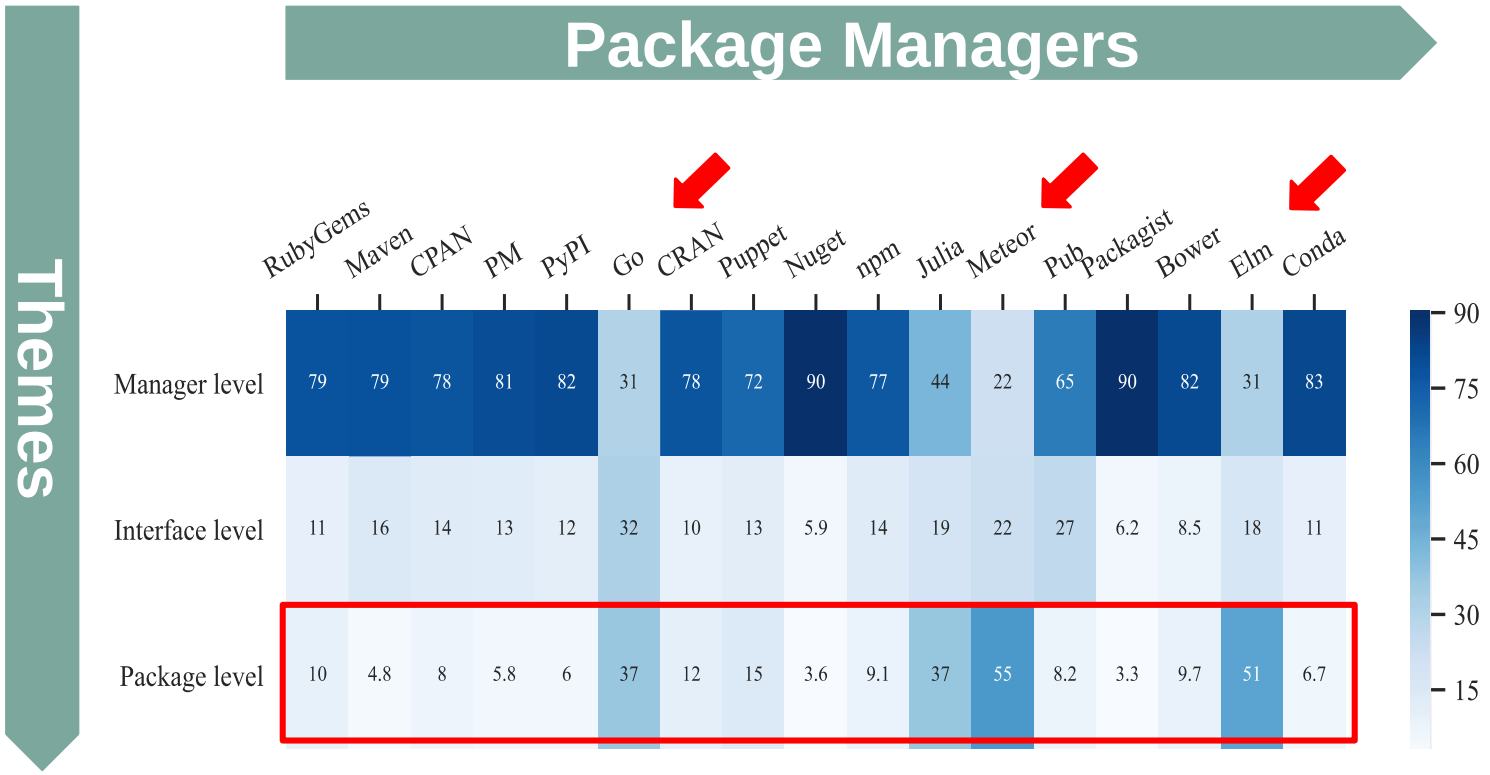
Packagist, Maven, CRAN users have dependency related issues

NuGet, CPAN, PyPI, Conda users have configuration related issues

RubyGems, Puppet users have error related issues

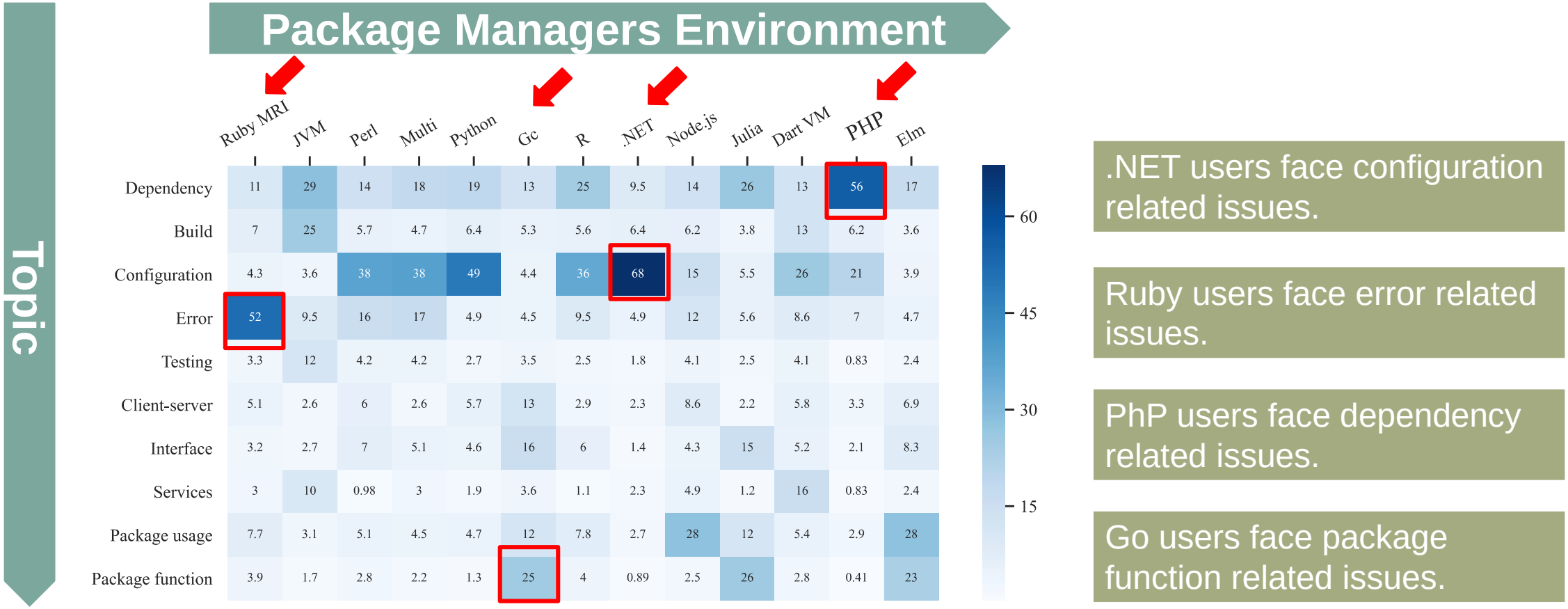
GO, Meteor, and Elm users have package usage and functionalties related issues

Contrasts in Topics: Themes and Package Managers [2/2]



GO, Meteor, and Elm users may face different issues compared to other PM users

Contrasts in Features: Environment and Topics [1/2]



.NET users face configuration related issues.

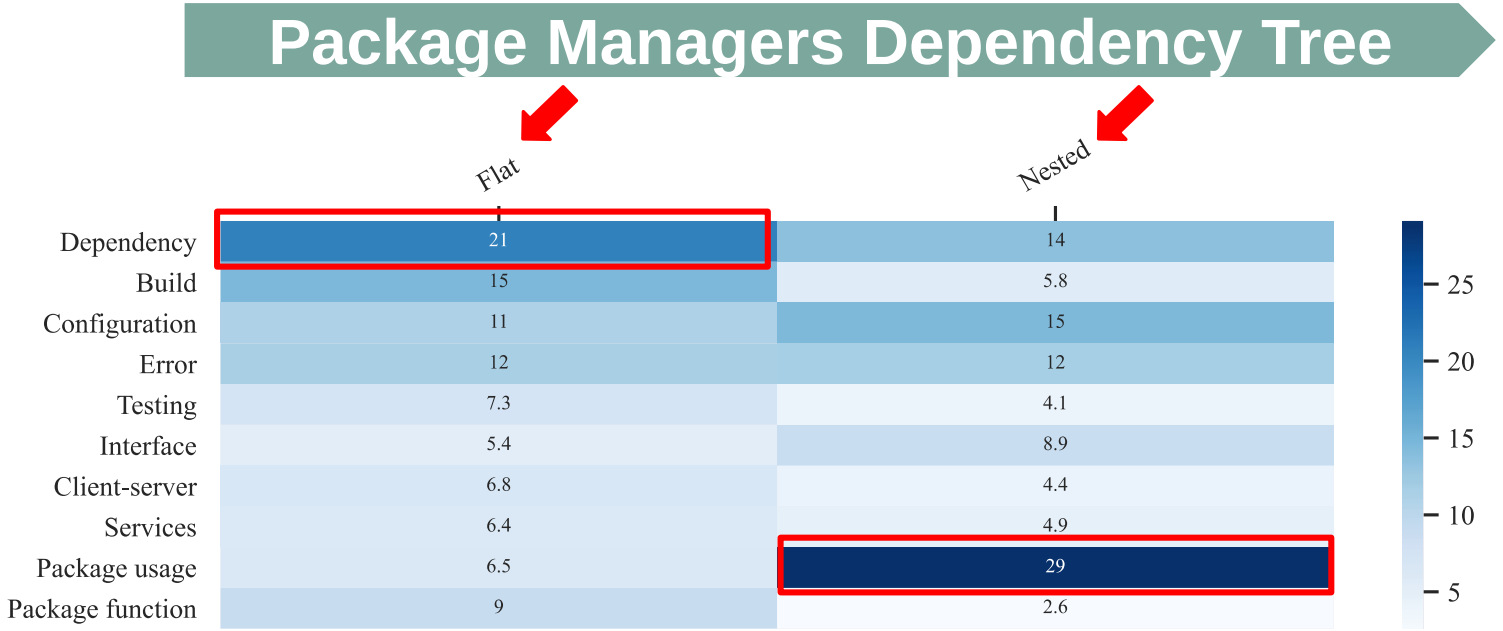
Ruby users face error related issues.

PhP users face dependency related issues.

Go users face package function related issues.

Contrasts in Features: Dependency Tree and Topics [2/2]

Topics



PM with flat dependency tree have high library dependency issues

PM with nested dependency tree have high package usage issues

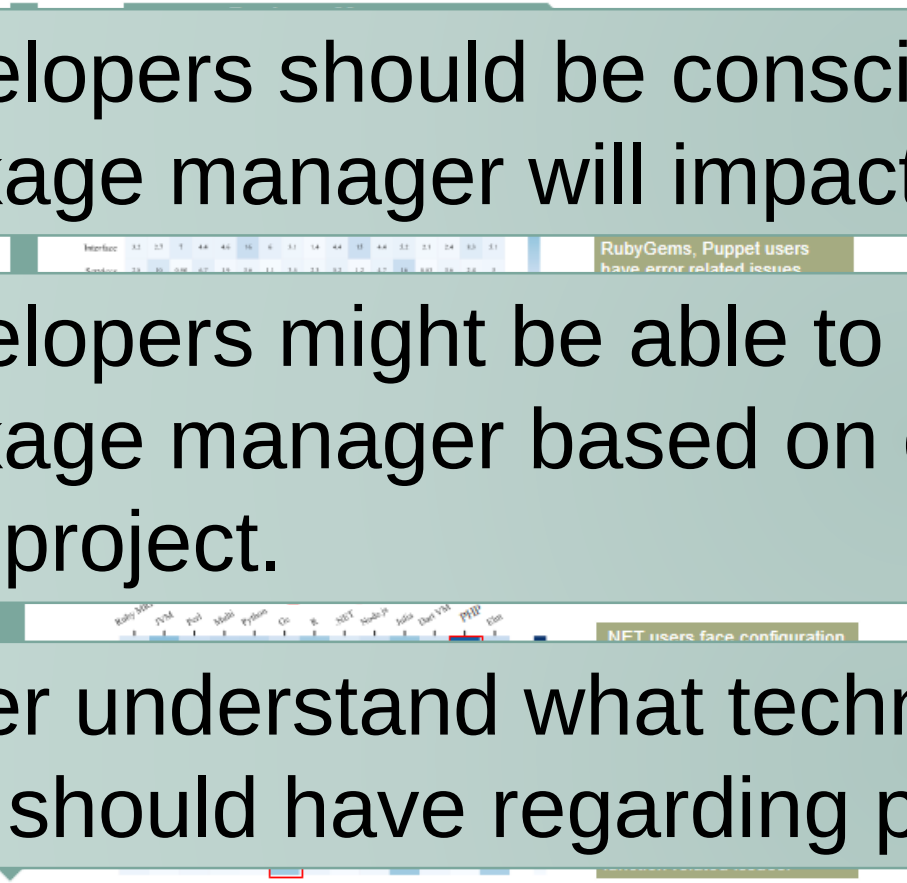
Contrasts in Response: Popularity and Difficulty

Theme	Topics	#question	#Score	#Views	#Favorite	Accepted Ans. (Avg.)	PD Score
Package Manager Management	Dependency	42,458	1	404	0	0.49	0.25
	Build	27,541	1	405	0	0.46	0.25
	Configuration	27,273	1	342	1	0.45	0.29
	Error	26,193	1	438	0	0.44	0.23
	Testing	14,404	1	434	0	0.45	0.23
I/O	Client-server	14,043	1	314	0	0.48	0.32
	Interface	13,673	1	265	0	0.54	0.38
	Service	13,442	1	402	0	0.43	0.25
Package	Package usage	27,422	0	215	0	0.53	0.47
	Package function	16,321	1	255	0	0.66	0.39

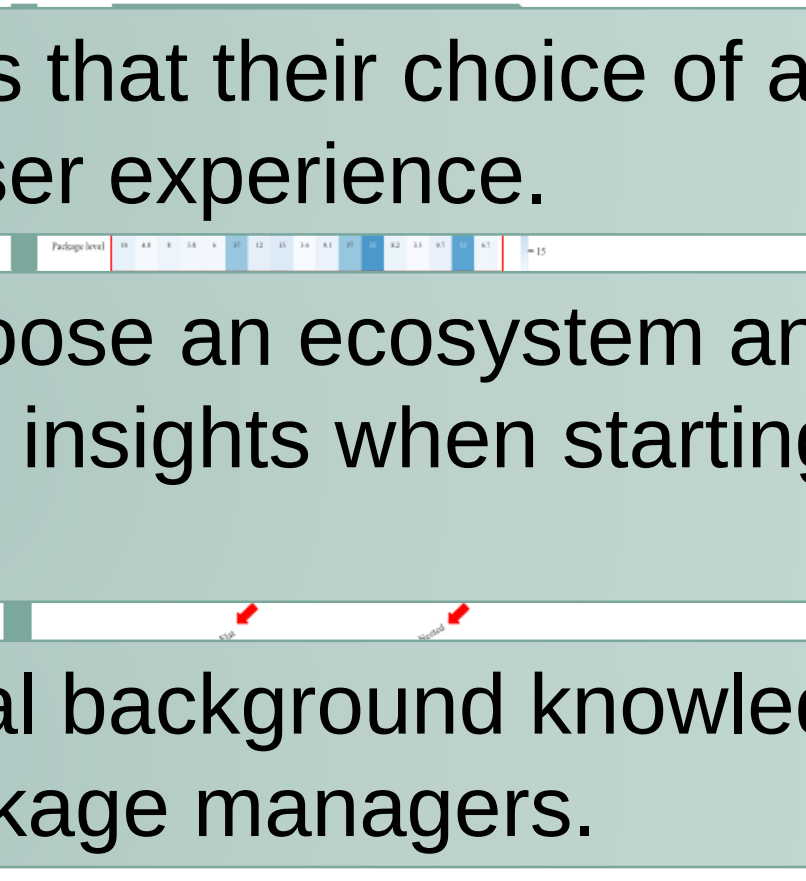
Go, Elm, and Meteor may have relatively easy experience

Implications for Developers [1/3]

Contrasts in Topics:
Topics and Package Managers [1/2]



Contrasts in Topics:
Themes and Package Managers [2/2]



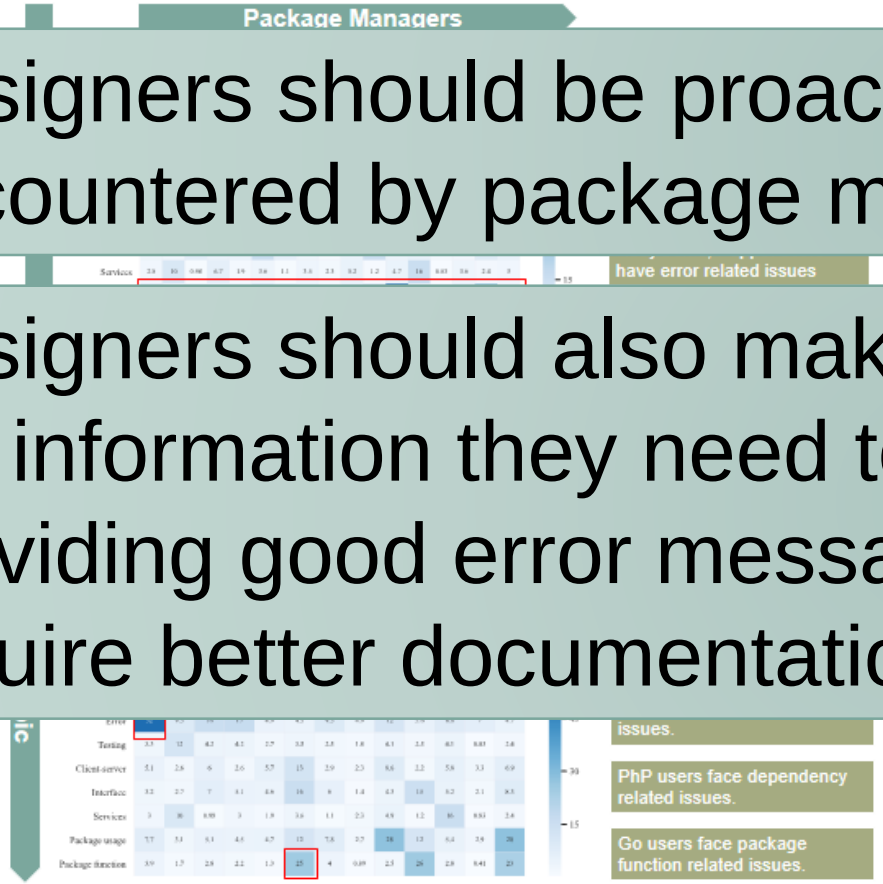
Developers should be conscious that their choice of a package manager will impact user experience.

Developers might be able to choose an ecosystem and package manager based on our insights when starting a new project.

Better understand what technical background knowledge they should have regarding package managers.

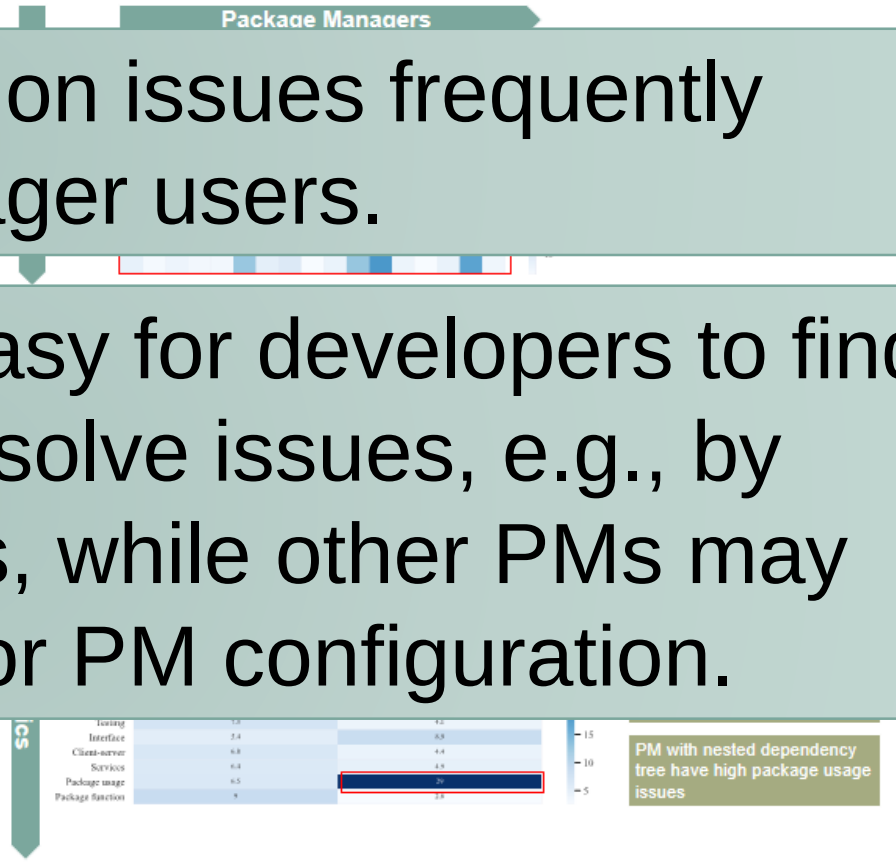
Implications for Package Manager Designers [2/3]

Contrasts in Topics:
Topics and Package Managers [1/2]



SoHeal@2021 16

Contrasts in Topics:
Themes and Package Managers [2/2]



SoHeal@2021 17

Designers should be proactive on issues frequently encountered by package manager users.

Designers should also make easy for developers to find the information they need to resolve issues, e.g., by providing good error messages, while other PMs may require better documentation for PM configuration.

Implications for Researchers [3/3]

Researchers can investigate the trade-offs between design features and potential issues to understand what an ideal package manager would look like.

Researchers could use our findings to prioritize research efforts, as our work is the first to acknowledge that developers encounter issues when using PMs.

Conclusions

- We studied 222,770 Stack Overflow Posts to investigate 16 package manager users experience.
  - Analysis results indicate that specific features of Package Managers are correlated with users experience.

Developers should be conscious that their choice of a Package Manager will impact experience

Designers should be proactive on the issues frequently faced by package manager users.

Research can investigate the trade-offs between design features and potential issues.