

Semi-structured interviews:

add at end of each cluster of questions “Are you happy with ... ? If not, why?”

Creation of configuration options

- (1) For which purposes you create new configuration options?
- (2) Why people create new configuration options? options-as-commands vs. optimization-of-performance vs. ...
- (3) Who decide to add a new option? Politique? Directives? Ad hoc? Clean-up/monitoring of which options are used?
- (4) Who are involved in the creation of a configuration option? And what is the role of each of these involved people?
- (5) How do they communicate? By using which artefacts (mails, ...)
- (6) Are there explicit requirements for a new configuration option?
- (7) Do options have any best practices? like config names should follow a certain pattern, they should be documented in a certain way ...
- (8) Do you use 3rd party frameworks, your own, ...? Do you know any best practices used to access a configuration option in the source code? For example, using a popular configuration framework, or a specific method, ... In the source code, how a developer can distinguish between options and ordinary variables or data inputs?
- (9) What kind of documentations you provide for configuration options?
- (10) How do you decide a configuration option default value?
- (11) How are option dependencies communicated to end-users?
- (12) What other kind of problems do you face during the creation of options? And how do you avoid/resolve them?
- (13) How do you define the difference between a configuration and an ordinary input?
- (14) Does user feedback has an impact on option creations/modifications?

Quality assurance

- (15) How do you prevent configuration errors?
 - (a) By executing automated tests, functional tests?
 - (b) By improving documentations? How?
 - (c) By improving error messages? How?
 - (d) By developing additional exceptions to deal with configuration errors?
 - (e) By proposing an automatic solution that help users resolving their bugs? Which ones?
 - (f) By storing configuration errors and their misconfigured options in a dedicated configuration errors database?
 - (g) Do code reviewers care about configuration options?
 - (h) If yes, what kind of feedback related to configuration options do they give?
- (16) Are configuration errors treated the same as regular errors? Same repo? Same importance? Same developers fixing them? ...
- (17) When a user report a configuration error:
 - (a) How do you know which option he has to change to fix the error?
 - (b) How do you reproduce a reported error?

- (c) How do you know the correct value he has to use to fix the issue?
- (d) How difficult it is to find the misconfigured option, and the correct value?
- (18) How difficult is it limiting the errors that can be introduced by other system dependencies? I.e., testing of others' options?
- (19) Tests updated with user feedback about erroneous options?

Maintenance of configuration

- (20) What is the percentage of dead options in the system you were involved in its development?
- (21) How do you deal with dead options? Do you remove them once they are not used anymore? After few version? Why developers do not immediately remove dead options?
- (22) Do you consider any refactoring of options? How often? What kind of refactorings (change the name of an option, remove an option, add a new document ...)
- (23) What could be the side effects/risks of refactoring configuration options? Does that stop you from refactoring?
- (24) What other problems do you face in maintaining configurations?

Users' perspectives

- (25) How can you characterise the complexity of a configuration option?/What kinds of options perceived to be difficult to use by users?
 - (a) By the configuration type: key-value, category (example of htaccess), ...
 - (b) By option types: int - string - complex objects
 - (c) constraints/dependencies between options
 - (d) lack of documentation
 - (e) unclear name
 - (f) Which of these categories are the best (easier to debug and fix in case of errors)
- (26) Are you aware/do you measure which options are being used by users? If so, what do you do with that information?
- (27) What are user feedbacks on new options?
- (28) What are user feedbacks on changes to existing options?
- (29) Are there any hidden/unofficial options? Is that a good or bad practice? Why?
- (30) How often users have problems with the default value?
- (31) Do you have configuration values that are automatically set depending on the platform and environment in which a system is executed? How do you decide about which ones are done automatically?
- (32) Do you consider in the development of a system the configuration of components that system depends on? Or is that information hidden from end users?