Survey on Multi-language Practices

Thank you for agreeing to participate, it will only take 15 minutes to complete.

Study Policy:

- Participation in this study is completely voluntary. If you decide not to participate there will not be any negative consequences. If you decide to participate, you may stop participating at any time and withdraw entirely your participation or you may decide not to answer any specific question.
- Your identity and the data collected thanks your participation will remain anonymous and will never be released to the public. Only anonymous data (aggregated or not) will be published in scientific articles, ensuring that the data cannot be linked back to a particular participant. The data will be kept by the principal investigator for five years before being destroyed.
- By submitting this survey, you are indicating that you have read the description of the study, are over the age of 18, and that you agree to the terms as described.

If you have any questions or would like a copy of this consent letter, please contact us at mouna.abidi@polymtl.ca

Study Design:

This survey is conducted by the **Ptidej research team** from Polytechnique Montréal and Concordia University in Canada.
The purpose of this study is to **investigate the practices used when dealing with multi-language systems**. Systems developed using more than one programming language. We aim to investigate what are the challenges and how do developers deal with such systems. Our main goal is to improve the quality of those systems.

Through this survey we aim to answer the following research questions:

- What benefits do developers perceive with multi-language programming?
- What challenges do developers face in multi-language systems?
- What practices do developers use in multi-language systems to overcome these challenges?
- How are those practices perceived by developers?

Definition of terminologies:

(Khomh, F., & Gueheneuce, Y. G. (2008, April). Do design patterns impact software quality positively?. In Software Maintenance and Reengineering, 2008. CSMR 2008. 12th European Conference on (pp. 274-278). IEEE.)

- Expandability: The degree to which the design of a system can be extended.
- Simplicity: The degree to which the design of a system can be understood easily.
- Reusability: The degree to which a piece of design can be reused in another design.
- Learnability: The degree to which the code source of a system is easy to learn.
- Understandability: The degree to which the code source can be understood easily.
- Modularity: The degree to which the implementation of the functions of a system is independent of one another.

Please feel free to share this survey with your contacts who have experience with multi-language development.

Thank you,

Best regards,
Mouna Abidi

Page 1 - Section I: Background

Section I: Background

*** 1. What is your role within your organization?**

(Yamashita, A., & Moonen, L. (2013, October). Do developers care about code smells? an exploratory survey. In Reverse Engineering (WCRE), 2013 20th Working Conference on (pp. 242-251). IEEE.)

- Software Engineer
- Team Lead
- Architect
- Project Manager
- Other, please specify
- Developer
- Tester
- QA Manager
- Self-employed

*** 2. How many years of experience do you have in software engineering?**

- Less than 1 year
- 1 year - 5 Years
- 5-10 Years
- More than 10 years

*** 3. What is the domain of activity of your organization?**
(https://en.wikipedia.org/wiki/Outline_of_software_engineering)

- Research and development
- Networks
- Healthcare
- Analytics (Business,IT services, BigData...)
- Banking and insurance
- Robotics and Embeeded systems
- Games
- Other, please specify

*** 4. What is your level of skill in the following languages? Please specify which other languages if relevant:**
(https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages)

	1 Novice	2 Little Knowledge	3 Practical	4 Comfortable	5 Expert
Python					
C++					
Java					
C					
C#					
Php					
R					
JavaScript					
Other, please specify					

5. Rank the following programming paradigms according to how familiar you are with each?
(https://www.cs.bham.ac.uk/research/projects/poplog/paradigms_lectures/lecture1.html)

	1 Novice	2 Little Knowledge	3 Practical	4 Comfortable	5 Expert
Imperative					
Functional					
Logic					
Object Oriented					

Page 2 - Section II: Multi-language

Section II: Multi-language

*** 6. Have you heard/used multi-language programming** (programs developped using more than one programming language)**?**

- Yes
- No

*** 7. To which domain your last multi-language project belongs to?**

- Client/Server Application
- Desktop Application
- Mobile Application
- Embedded Systems/Operating Systems
- Other, please specify

* 8. Which pairs/sets of programming languages did you use when you developed multi-language project(s)?

* 9. In your opinion, has the use of multi-language programming increased or decreased over time?

- Increased over time
- Decreased over time
- Remained the same
- Impossible to know

10. From your previous answer, why did the use of multi-language increase or decrease over time?

* 11. From your experience, how do you evaluate the impact of using more than one programming language in a project, to the following aspects of software development?

Mayer, P., Kirsch, M., & Le, M. A. (2017). On multi-language software development, cross-language links and accompanying tools: A survey of professional software developers. Journal of Software Engineering Research and Development, 5(1), 1.

	Negative impact	Neutral	Positive impact	N/A
System Performance				
System architecture creation				
Implementation of initial code				
Memory usage				
Understandability of the system				
Translation of requirements to code				
Motivation of developers				

* 12. What are the main issues you have encountered in your multi-language project(s)?

- Security
- Performance
- Correctness
- Maintenance
- Robustness
- Other, please specify
- None of the above

13. How did you solve those issues?

14. Do you use any practices or patterns when developping multi-language systems?

* 15. Do you have any experience or heard about JNI (Java/C(++)) Development?

- Yes
- No

- Go to **page 4 - Section IV: Patterns and practices** if
  15. Do you have any experience or heard about JNI (Java/C(++)) Development?
  is *No*

Page 3 - Section III: JNI

Section III: Java Native Interface (JNI)

*** 16 How often do you use JNI in your projects?**

- Very Often
- Often
- Rarely

17. From your experience, what are the advantages and disadvatanges of using JNI?

Advantages:

Disadvantages:

*** 18 Do you prefer using primitive or reference types when using JNI?**

- Primitive than reference types
- Reference than primitive types
- No preference

19. From your previous answer, why would you prefer reference/primitive types?

*** 20 How often do you use the following in your JNI projects?**

	1 Very often	2 Often	3 Rarely	N/A
Using field access and method callbacks	○	○	○	○
Use of exceptions	○	○	○	○
Use of native threads	○	○	○	○
Use of primitives (GetStaticFieldID and GetFieldID)	○	○	○	○
Other, please specify	○	○	○	○

Page 4 - Section IV: Patterns and practices

Section IV: Patterns and practices

*** 21 Which learning sources do you use to learn multi-language patterns and practices?**

- Blogs
- Discussion forms
- Programming Language Specification (example: JNI Specification)
- Books
- Articles
- Stack Overflow
- Youtube/Video tutorials
- Other, please specify

*** 22 Are you following any of the below practices when developing multi-language systems?**
(https://www.ibm.com/developerworks/library/j-jni/index.html), (https://developer.android.com/training/articles/perf-jni)

	1	2	3	N/A

	Very Often	Often	Rarely	
Check Exceptions				
Check multi-language return values				
Take care of and release the strings				
Avoid asynchronous communication between the languages				
Minimize the number of threads that need to touch or be touched by the languages				
Safe load of the library (example: AccessController)				
Other, please specify				

*** 23 How do you evaluate the impact of the following practices on those quality attributes?**

Select where you believe there is a positive impact
(https://www.ibm.com/developerworks/library/j-jni/index.html), (https://developer.android.com/training/articles/perf-jni)
(Khomh, F., & Gueheneuce, Y. G. (2008, April). Do design patterns impact software quality positively?. In Software Maintenance and Reengineering, 2008. CSMR 2008. 12th European Conference on (pp. 274-278). IEEE.)

	1 Expandability	2 Simplicity	3 Reusability	4 Learnability	6 Understandability	7 Modularity	N/A
Check Exceptions							
Check multi-language return values							
Take care of and release the strings							
Avoid asynchronous communication between the languages							
Minimize the number of threads that need to touch or be touched by the languages							
Safe load of the library (example: AccessController)							
Other, please specify							

24. If you answer N/A in the previous question, please provide an explanation

- Lack of knowledge about patterns and practices
- Lack of interest on patterns and practices
- Lack of time
- Other, please specify

*** 25 How often do you encounter the following pitfalls in your project(s)?** (https://www.ibm.com/developerworks/library/j-jni/index.html),
(Tan, G., & Croft, J. (2008, July). An Empirical Security Study of the Native Code in the JDK. In Usenix Security Symposium (pp. 365-378))

	1 Very Often	2 Often	3 Rarely	N/A
Not caching method IDs, field IDs, and classes				
Using many local references without informing the JVM				
Not checking for exceptions				
Not checking return values				
Using global references incorrectly				
Buffer overflows				
Memory Management flaws				
Other, please specify				

*** 26 How do you evaluate the impact of the following pitfalls on those quality attributes?**

Select where you believe there is a negative impact

	1	2	3	4	6	7	N/A

	Expandability	Simplicity	Reusability	Learnability	Understandability	Modularity	
Not caching method IDs, field IDs, and classes	☐	☐	☐	☐	☐	☐	○
Using many local references without informing the JVM	☐	☐	☐	☐	☐	☐	○
Not checking for exceptions	☐	☐	☐	☐	☐	☐	○
Not checking return values	☐	☐	☐	☐	☐	☐	○
Using global references incorrectly	☐	☐	☐	☐	☐	☐	○
Buffer overflows	☐	☐	☐	☐	☐	☐	○
Memory Management flaws	☐	☐	☐	☐	☐	☐	○
Other, please specify	☐	☐	☐	☐	☐	☐	○

27. Are you using any tools or methods to detect multi-language patterns and good/bad practices? If yes please describe.

*** 28 How do you verify that you have avoided the common pitfalls when using multi-language systems?**
(https://www.ibm.com/developerworks/library/j-jni/index.html)

- ☐ Validation of the code against specification (Checking exceptions, return values...)
- ☐ Analyzing the method trace
- ☐ Using dedicated tools
- ☐ Performing code reviews
- ☐ Generating dumps
- ☐ Other, please specify

*** 29 How much do you refactor to improve your multi-language code and remove bad practices?**

- ○ Very Often
- ○ Often
- ○ Rarely
- ○ Never

*** 30 From your experience, how do you evaluate the effort needed to remove bad practices of multi-language systems in those phases?**
(https://www.linkedin.com/pulse/what-software-development-life-cycle-sdlc-phases-private-limited/)

	1 Low effort	2 Medium effort	3 High effort
Requirement gathering and analysis	○	○	○
Design	○	○	○
Implementation or coding	○	○	○
Testing	○	○	○
Deployment	○	○	○
Maintenance	○	○	○

31. Could we contact you for a short interview by phone, Skype, or e-mail? We guarantee your anonymity and that of your answers. If yes, please add your e-mail to contact you.

Your responses have been registered!

Thank you for taking the time to complete the survey, your input is valuable to us.