Assignment - Devops Knowledge

> Submission for `Aditya Girisha (012145901)`

Introduction

- This assignment is a read-through of the following paper:
  - `Automated Capturing and Systematic Usage of DevOps Knowledge for Cloud Applications`
  - Autored by Johannes Wettinger, Vasilios Andrikopoulos, Frank Leymann
- The objective of the assignment is to go through the paper and answer the provided questions.

Question 1

What do you believe were the key contributions of this paper?

- `The Introduction` highlights the need for a good operational workflow to ensure continous integration and and continous delivery
  - CI/CD helps a product to apply patches and introduce new features rapidly.
  - The paper tries to comb through the various offering of DevOps Solutions (scripts/tools/frameworks) and tries to offer a solution (of sorts) to using these resources effectively
- It first goes through the numerous `variables that need to be considered` while trying to deploy your application, such as:
  - Configurational Independence
  - Effective Scaling
  - Providing Continous Delivery
  - Choosing the right tools
  - Continued Product Support
- It tries to introduce an `Holistic Approach` (ie, consider all possible options in a systematic way)
  - This is to ensure that users
    - Choose the right DevOps solutions
    - Design and Implement these solutions effectively

- The approach can be summarized with the following
    - Define the application
    - Analyze the DevOps requirments
    - Compile a Knowledgebase
    - Reference a knowledgebase for solutions/designing/prototyping

- A `Knowledge Base (KB)` is a repository of information, typically used to store, organize, and manage knowledge. A KB can contain:

    - Dedicated Blogs, FAQs, KnowledgeStacks (ex: stackOverflow.com, devops.stackexchange)
    - Subject Matter Experts (ex: Consultants, IEEE, ISO, etc)
    - Documentations (tool/framework documentations, publications, whitepapers)
    - Commercial Templates (AWS Marketplace, Cloud Formation templates etc)
    - Crawlers
        - Search Engines, LLMs, Articles

- The paper touches upon how to use this KB using the `Crawling Approach` which includes

    - Discovering Knowledge
    - Handling Different Sources of Knowledge (Structured and Unstrucutred)
    - Capturing Refined Knowledge
    - Populating knowledge effectively
    - Querying KB according to current requirements
    - Refining the design over multiple queries to KB
    - Keeping the KB updated over time

Question 2

What flaws did you see of the work described in this paper?

Some flaws that I feel need to be touched are:

- Overboarded Knowledge Management
    - Considering the scale of the product before creating and managing complex knowledge base
    - Does the organisation have enough time/resources to dedicate?
    - Does the organisation require extensive KB at initial stages
- Refinement Timeline
    - Will the usage of certain Operational Solutions restrict or delay the development of the product
    - Does the application need to be reworked to fit the solutiion

    - Does the Application face any degradation in performance or loss of features due to implementation of certain Devops Solutions
- Tight Coupling
    - Will the usage of certain Devops Solution make it coupled tightly with business solution
    - Can a particular solution be easily interchanged int the future?
    - If a devops solution undergoes changes in dependencies will our product still be functional?
    - Can a Devops solution integrate easily with existing application
- Pricing
    - Licensing Cost for Devops Solution
    - Cost for Literature Survey and Maintaing knowledge base
    - Initial Infrastructure Costs
    - Cost of hiring Subject Matter Experts
    - Cost of training staff on new tool

Question 3

What are the logical next steps for this work or what topics should have been explored further?

- Advanced Crawling Techniques
    - Using AI/ML to discover and refine data (with proofchecking ofcourse)
    - Crawling techniques should factor in technical debt and coupling
    - Crawling techniques should factor in price
    - Crawl Discovery should research industry implementations
- Advanced Knowledge Management Techniques
    - Creating affinites relations between various solution
        - Ex : Docker and Kubernetes / AWS and Cloudformation
    - Tagging Knowledge Discovery with areas of interest
        - Ex : Spring is Monolithic structured and docker is microservice structured
- Defining effective Knowledge base
    - Readability
    - Traversability
    - Reliability (up to date on correct information)

Question 4

Anything else about the paper?

- I was intrigued by `Section V : Knowledge base Utilization` which assign predicates to product requirements
- While this section shows a academic approach of detemining the minimum requirements as seen in the equation mention

```
WordPress_minimum =
P(‘Middleware/DB/.../MySQL’, ‘versions’, ‘5.0’) ∧
P(‘Middleware/Runtime/PHP’, ‘versions’, ‘5.2.4’) ∧
P_requires(‘Middleware/Web Server’)
```

- This kind of mathematical calclulations are not usually seen within a live environment (atleast from my time working as a DevOps consultant)
- Requirements are usually represented in cloud diagrams where interconnectedness is more apparent
- Version compatability is usually consolidated in prototypes to maximize usage of LTM candidates or latest versions for underlying dependencies
- Such a rigid formula is something I dont think a lot of people are using on a day to day basis