

## TOWARDS AFRICAN METAPHORS AND PROVERBS EXPERT SYSTEMS FOR INTERNATIONAL DEVELOPMENT PROJECTS

Wangai Njoroge Mambo<sup>1</sup>, <https://orcid.org/0000-0002-6496-2159>

<sup>1</sup>United States International University–Africa, Nairobi, Kenya

### ABSTRACT

Some participation and collaboration problems between local beneficiaries and foreign experts in international development projects (IDPs) are due to two groups using different types of knowledge: indigenous and global respectively. Study proposes creating locally low cost indigenous knowledge expert systems and startups incrementally to solve participation and collaboration problems that can be evolved and scaled up in a sustainable way. Method nexus for evolutionary prototyping product and startup creation is explored. Future IDPs will be supported by artificial intelligence enabled tools requiring some basic local AI understanding. Complementary indigenous knowledge expert systems can create synergy with IDPs artificial intelligence systems but will require local ES development capabilities to be built and evolved in advance.

**Keywords:** international development projects, indigenous expert systems in Africa, indigenous techniques.

---

Manuscript first received: 2024-01-01. Manuscript accepted: 2025-12-01

*Address for correspondence:*

Wangai Njoroge Mambo, United States International University–Africa, Nairobi, Kenya  
E-mail: [mambon20@aol.com](mailto:mambon20@aol.com)

## INTRODUCTION

International development projects (IDPs) sometimes fail by not integrating local community knowledge. Indigenous knowledge makes projects understandable, easier to adopt and make sense to local people. It's difficult to completely embrace what one does understand well. Proverbs represent local understanding of things, are a language of thought for mobilizing local insight and visualizing issues through indigenous points of view (Local researchers and Easton 2004), they create maps for navigating novel phenomena. Proverbs link current issues with past local innovation experiences and future aspirations within cultural framework (Local researchers and Easton 2004). Synthesizing proverbs with global knowledge can open new frontiers like creating much needed indigenous innovation techniques (Mambo, 2021). For example creating and adapting artificial intelligence (AI) research, innovation and development techniques appropriate for local context. IDPs effective learning process should build on local people indigenous knowledge. Locals and non-locals can use proverbs to better understand each other; build confidence and share knowledge.

AI helps improve lives, reduce inequalities, is changing societies similar to steam engine and electricity (Cisse, 2018). Electricity light cities, towns and villages but AI metaphorically provides more powerful lighting for the mind and new eyes. Africa failed to join the first, second industrial revolutions, partially joined the third revolution mobile phone services segment. The success of mobile services revolution was due building on previous locally developed manual solutions and collaboration between local and international telecommunication and financial firms in developing automated locally appropriate solutions founded on local culture. AI especially expert systems (ESs) can enable local people leverage and build their expertise more efficiently and effectively. Creating simple partial but usable AI knowledge based solutions containing local proverbs and metaphors can be a starting point.

ESs provides expert advice to users by processing knowledge. ES architecture consists of knowledge base, inference engine, user interface, knowledge explanation and acquisition modules and relationships between them. Explanation module explains ES reasoning that producing results. Knowledge acquisition module assists in eliciting knowledge from the domain experts. In study's ES projects domain experts are local stakeholders who possess proverb expertise and local and foreign experts who possess development project knowledge. A knowledge base consists of knowledge elements represented as facts and rules that are used by inference engine to advice and answers user queries. Knowledge engineers are responsible for capturing domain expert knowledge and creating knowledge base.

The study's objective was to find means to create ESs in continuous small increments to sustainably build local ES capabilities to support IDPs. Future IDPs projects will require Local AI capability to create local support tools that interoperate with donors AI enabled computing solutions. This will require beneficiaries to have some basic AI understanding and local computing practitioners to have AI building capabilities.

## **RESEARCH METHOD**

This study used design science research method (DSRM) used to develop expert systems (Miah and Genemo 2016), software engineering and information systems (Wieringa, 2014). DSRMs support innovation, research and development activities (Hevner, March, Park and Ram 2004). The DSRM method selected has been adapted to research, innovation and development (Vaishavi and Kuecher 2009). Research followed DSRM steps: awareness of problem step identified problem of local development project beneficiaries understanding and participating in projects from published literature, suggested solution step proposed exploring ES development. Development step created method nexus through evolutionary prototyping and knowledge base prototype. DSRM aids in navigating ES creation landscape maze of forces, challenges and scarce resource constraints.

## **DISTANCES AND INDIGENOUS KNOWLEDGE**

Seven distances between and across local beneficiaries, local and foreign experts working together are: cognitive, communicational, organizational, cultural, functional, social and geographical presenting opportunities for innovation (Parjanen, 2008) but also challenges that make cooperation and collaboration difficult. Cultural differences are due to every organization and its subunits viewing knowledge differently by having different habits, rules and values (Parjanen, 2008). The differences also occur between and across formal and informal organizations. The three types of IDP stakeholders have different types of expertise and knowledge that creates functional distance. Communication distances make it difficult to learn and understand each other. Knowledge depth and breadth differences create cognitive distance. Distances form a maze landscape with opportunities that can be exploited through innovation and challenges that create project threats.

The distances aggregate at higher levels creating national culture and disciplinary differences. Local experts will often have same national culture as project's local beneficiaries but have formal and informal organizational cultures. Local and foreign experts from different disciplines have different disciplinary cultures. Unless local and foreign experts work for same organization which is often rare they have different organizational cultures.

Proverbs enable development of technology solutions for African people by African researchers applying proverbs in all stages of research process (Iyamu, 2021). If participatory design science research methods are used they would enable indigenous innovation solutions development through research. These methods can be adapted to fit traditional way of making everyone a knowledge consumer, creator and innovator (Mambo, 2021). There are many problems requiring solutions than global knowledge experts don't have time to solve and participation by everyone is necessary if more problems are to be solved.

Proverbs can bridge distances by enabling experimentation and learning from doing that can be automated with AI. Metaphors can help local people understand science and technology, community of scientists and technologists, for example understanding scientific inquiry as learning a foreign

language (Harwood, Reiff and Phillipson 2005). Local people can invent jigsaw puzzle ideas from metaphors like learning language that they assemble to visualize global phenomena. Designing jigsaw puzzle with proverbs and metaphors pieces, assembling them and solving the puzzle (Chang, 2004) is approach to applying indigenous solutions to solving IDPs problems.

## **SCARCITY OF ARTIFICIAL INTELLIGENCE EXPERTISE**

While African AI research and development experience goes back several decades, it thinly distributed over large areas in the continent. There few areas in the continent where it's possible to carry out medium scale AI projects. Most developing countries cannot afford to build AI expertise in all its domains but can select one or few AI domains based criteria likelihood of succeeding, ease of starting, availability of initial expertise, probability of scaling in a low cost way, creating a domain strategy and later move to other domains. Two candidate starting domains are machine learning and ESs. Machine learning domain is globally fastest growing while ESs are AI systems best suited to assist novices upgrade their capabilities.

Expert systems based systems (ESBSs) are classical software systems with one or a few expert system components. Replacing classical software components with ES components is innovation. There are four types of innovations: modular innovation substitutes classical software components with ES components and retains the relationships, architectural innovation introduces new relationships between ES modules, radical innovation introduces ES components based new knowledge and new relationships between them and incremental innovation makes small improvements to knowledge of ES modules and relationships between them (Henderson and Clark 1990). The four types of innovations can be at module, subsystem and system levels. Module incremental innovation makes small improvements to module elements knowledge (program instructions and blocks) and relationship between the elements. An ESBS with a few ES modules is modular innovation at sub system level. Once there is enough expertise and understanding research and innovation it can drive ESBS evolution. Design science methods can be adapted for catch-up model ES development to make them suitable for Africa context.

Similarity of ESs with classical software makes them likely to benefit more from research and practices of applying software engineering to artificial intelligence (SEAI) than other AI domains. ES analogy and analogy inspired design can improved by learning from bioinspired design by applying SEAI experience.

## **CREATING LOW COST EXPERTS SYSTEMS INCREMENTALLY**

Major AI developments will solve real problems facing ordinary people using their ways (Nayebare, 2019), based on indigenous knowledge. Proverb transdisciplinarity (Mambo, 2021) can be used to synthesize global and indigenous knowledge to enhance local and foreign stakeholder's collaboration. Is it useful to use transdisciplinary AI research approaches in Africa? (Nayebare, 2019) Transdisciplinary ES and software engineering methods can be used to develop ESs (Mambo, 2022). Transdisciplinarity methods can be taught to non-computing professionals and practitioners to be able participate in AI projects enabling projects to get critical mass of developers.

Transdisciplinarity is based on idea of developing solutions once and using it in all disciplines (Mambo, 2022) informal indigenous knowledge disciplines and informal sector. Developing a software engineering AI transfield (Mambo, 2022) would contribute towards helping Africa leverage indigenous and global knowledge in ESBS application, development and to build capabilities. While many African countries have developed many innovation strategies for different technologies transdisciplinarity can aid in unifying them.

African sustainable ES and knowledge base creation critical success factors(CSFs) are: keeping required funding minimal, spreading it over time, project costs low, using transdisciplinary approaches that enable non-computing professions to participate in ES projects and novice developers to upgrade their skills. AI and ES are non-core aspects of IDPs but their importance will increase in future and become core as AI becomes more pervasive and ubiquitous. AI systems consistent with local values and culture can enable quantum leapfrog skipping some stages of development (Nayebare, 2019) by aligning global knowledge, local culture and dynamics. Leapfrogging specific technologies can be accelerated by generic technology like AI and require time before a country can leapfrog (Bhagavan 2001). African countries leapfrogged by skipping several legacy communication technologies to mobile phone technologies (Amuomo, 2017) and lessons learned can be applied in AI and other information technology leapfrogging.

ESs augment human intelligence, capabilities and alter the way we gather information and do things (Strusani and Hounbonon 2019). DPs local beneficiaries and informal sector have limited access to knowledge and information due to digital divide and are likely to benefit most from AI solutions as they can easily augment their intelligence and capabilities. Automating indigenous intelligence will also aid in better understanding of its nature and make easy to use.

Starting by creating a simple usable proverbs and metaphors knowledge base is possible way to start ESs development in resource constrained environments. The knowledge base may be populated with only facts and latter rules added, or incomplete but usable facts and rules. It can be implemented in free programming languages like CLIPS and Prolog. Prolog is AI logic based programming language used to write intelligent instructions for computers to follow to perform actions that are considered intelligent when performed by people. CLIPS is a rule based AI programming language that uses a knowledge base of concepts and fact list. According to AI Turing test there should be no difference between results produced by AI programs and human experts. Many AI programs are useful although they don't perform as well as expert. AI Student interns can be used create the knowledge base. The knowledge base created can be licensed as open source to be developed and used by several development projects.

A Knowledge base is a knowledge management system used to store knowledge used for inference intelligent reasoning. Starting by implementing a small part of system for example the knowledge base using a programming language that available for free lowers project costs.

New African businesses developing expert systems are likely to be formal and informal startups. Startups start developing small products and they then scale them up. ES formal startup projects should be small and informal startup projects even smaller. Formal startups visions can be to become regional technology leaders while informal sector startups visions are likely to become slow technology followers.

Developing countries ESs should be based on barefoot knowledge engineering approach to meet human sustainable development objectives (Riesco, Artaso, Belandia and Pickin 2014). Barefoot is based on creating lowest cost bare minimum ES that is beneficial and affordable. Most informal sector products are of this type. Developing AI applications in Africa is an emergency that should be dealt with immediately using use low cost development solutions. Starting immediately enables capability and knowledge building making it possible to address some problems and to benefit from global AI knowledge spillovers. Missing the artificial intelligence revolution will widen the gap between AI technology leaders and laggards making it many times more difficult to catch-up compared with first, second and third industrial revolutions.

A web knowledge base accessible through mobile phones is widely available. Some university students struggle to get research projects, creating and evolving ESs and knowledge bases can be made computing student research and development projects options. There may be past ESs projects gathering dust in university archives that can reused in creating new ESs if copyright issues can be resolved.

ES can be evolved to support knowledge management, decision making, and training. The more features added to ESs the more complex and expensive they become. To be technically, economically feasible, sustainable and affordable for Africa, ESs should be small, start with smallest usable increments, taking into account constrains.

## **EXPERT SYSTEM BASED SYSTEMS DEVELOPMENT METHOD NEXUS**

ESBSs are developed using a combination of ES and software engineering methods because they consist of expert system and classical software components. Design science methods (Miah and Genemo 2016), Agile methods (Perkusich et al. 2019) and exploratory programming (Kerry and Myers 2017) are transdisciplinary (Mambo, 2022) used to develop both ES and classical software. There few established organizations in Africa developing artificial intelligence systems and fewer for expert systems. Creating AI and ES local development organizations will most likely start with formation of formal and informal startups. AI researchers and practitioners pioneering development of ESBS are likely to do it as a side hustle in unregistered startups. Software startup methods should be included as part combination of methods for developing ESBS. Two commonly used software startup methods are lean startup method (Ries, 2011) and design thinking. Design thinking was integrated with extreme programming (DTP) by starting every extreme programming life cycle step with design thinking (Hirschfeld, Steinert and Lincke 2011). DTP is agile innovation method can be used to develop ESBS based on best practices of applying software engineering to AI. Learn startup method can be used for technology innovation, research learning, building capabilities through attitude that allows leveraging transformative ideas and learning from mistakes (Ávalos, Pérez-Escoda and Monge 2019) which are necessary ESBS developing catch-up firms. Design thinking and lean startup methods can support bridging different type's distances and ESBS innovations.

Design theories and design methods nexus is classical software innovation method (Pries-Heje and Baskerville 2008). Method nexus for developing ESBSs should combine methods for developing classical software and expert systems: design thinking, design science, exploratory programming, lean startup method and agile methods. The degree each method is used at system, subsystem and step levels in development process is discovered by experimentation. **Table 1** shows different

ways of combining methods in the nexus. Design thinking methods consider multiple alternatives in all stages while explanatory programming and lean startup method support experimentation. Using DTXP in ESBS development method nexus includes both extreme programming and design thinking. Lean startup method (Ries, 2011) and extreme programming support experimentation, learning from innovating and from doing. Software development life cycle consists of planning, analysis, design and implementation learn startup method and exploratory programming should be used to experiment and learn which most appropriate method for a step or task to be performed. Transdisciplinary software engineering for expert system (Mambo, 2022) can be used build the theories for method nexus.

**Table 1.** Different ways of combining ESSD methods nexus.

Methods combination	Explanation
Main method	Best method is used as main overall method and other methods for sub problems that they are better than it
Prioritizing methods	Methods prioritized based on nature of project and method to be used in each situation selected by experimentation starting with one having highest priority.
Initial and other step methods	Methods divided into two groups those for initial step and methods for other steps
Initial part and other part of step methods	Methods divided into two groups those for initial part of step and methods for remaining part of the step. For example DTXP uses DT for initial part of every step and XP for other part of the step
Methods selected based on practice required for task	Select method to apply to task based on its support for practice required. Example if task requires learning from discovery then best method that supports learning from discovery is selected

Extreme programming requires customer to be involved throughout the development process. Integrating project techniques with proverbs enables IDP beneficiaries to be intelligent active contributors to ESBS projects. DXTP method was adapted with proverbs and presented in software pattern format (Mambo, 2017) can enable active participation beneficiaries. Applying proverbs based on pattern theory will enable integrating proverbs with patterns from other disciplines and domains. Software startup patterns (Cuckier and Kon 2015) provide widely applied software startup formation and evolution best practices that can aid in formation of informal startup and learning how build ESBS capabilities.

Learning from methods is software startup pattern (Cuckier and Kon 2015) that guides on learning how to create startups and products from scratch from business, engineering, computing disciplines and domains methods. This advice is especially useful for developing countries startups focusing on building product and startup development capabilities from scratch through experimentation. Technical debt pattern (Cuckier and Kon 2015) is best practice on how some important software and startup factors can be traded off to complete product development or creating startup operations necessary for startup survival.

## EXPERT SYSTEM PROTOTYPE

Prototypes are used to learn, understand requirements, improve communication, prove feasibility and to discover invention pathways. Design thinking allow creating several different types of prototypes to explore the entire design space and learn how to design systems (Brenner, Uebernickel and Abrell 2016). DTXP method can integrate these types of prototype.

Users query ES by using objects, metaphors, patterns and proverbs or formulate their queries. Global knowledge needed by participants should be adapted and represented using indigenous techniques and GK: techniques like innovation principles , design science patterns and others. Design science patterns capture widely used technology and management experience in a reusable form (Vaishavi, and Kuecher 2015). Analogy and abstraction was used to represent software development knowledge required by software engineers (Zhuge, Ma and Shi 1997). Both metaphors and analogy reason using source and target object attributes and relations (Hey, Linsey, Agogino and Wood 2008). Proverbs use abstraction to represent knowledge and experience. Metaphors and proverbs have the necessary elements needed to represent solution development knowledge and experience. Users should be asked after ES interaction session what they needed but couldn't locate or didn't exist.

Some design science patterns for example design and conquer are both patterns and proverbs. Software patterns, problem solving, indigenous technology invention and AI search are guided by graphs can be constructed using proverbs, innovation principles, sayings (Mambo, 2020), software and design science patterns. Experimenting with these graph can be used to discover ways of mastering global knowledge needed by participants to make informed decision to accept or reject projects. Knowledge base should represent these graphs and other techniques capable useful for testing participant hypothesis of understanding and applying global knowledge. A proven way of mastering global knowledge for people lacking global knowledge is to use indigenous knowledge to construct local paths to global knowledge (World Bank, 2004). Proverbs and metaphors are elements for constructing paths for IDPs mastering and path construction process can be supported by ESs.

The following are knowledge base prototype sample facts representing proverbs, metaphors, sayings, innovation principles, software startup and design science patterns heuristics that can be implemented in AI programming language like CLIPs:

```
(heuristic, general applications, project applications, project related proverbs).
(heuristic, substitutable concept, substituting concept)
(heuristic, metaphor)
(metaphorname, source object, target object)
(metaphorname, similar, sourceattribute1, target attribute1)
(metaphorname, similar, sourceattribute2, target attribute2)
.....
(metaphorname, similar, sourceattributen, target attributen)
(metaphorname, similar, sourcerelation1, target relation1)
(metaphorname, similar, sourcerelation2, target relation2)
.....
(metaphorname, similar, sourcerelationn, targetrelationn)
///
```

Proverbs information that is currently difficult to implement or its developers don't yet know how to implement efficiently in knowledge based is implemented in a database system. The database can consist of applicable proverbs and how they are applied in IDPs projects as well as beneficiary experts who can be consulted on how to apply proverb. The goal is to gradually evolve data and database structures into knowledge base knowledge and structures when knowledge and capability becomes available.

Developing small ESBS by experimenting allows small wins from the beginning that motivates further experimentation, increasing likelihood of funding, makes it easier to correct and learn from mistakes as the components are small. Learning from mistakes ensures similar mistakes are not made in future increments.

## CONCLUSION

Low cost bare minimum self-boot strapping ES development strategy makes them compete better for limited resources such as funding and increases chances of ES projects succeeding. These create opportunities for ES leapfrogging.

Starting by creating smallest and simplest usable knowledge bases and scaling them up in directions of increasing usefulness is a bare minimum approach of getting started quickly, sustaining ES development and capability building. These knowledge bases are easier to fund as they require funds in small amounts and not upfront. This can increase number of expert systems developed and if several of these projects succeed more funding is likely to become available.

Method nexus be evolutionary prototyped through experience gained during building ESBS. Applying artificial intelligence to automate indigenous sector must involve automating indigenous knowledge. Empowering majority of people with AI tools can increase productivity and enable reaching success tipping point.

## REFERENCES

- Amuomo, N. (2017). Absolute Poverty and the Desire To Emancipate Communities Driving Mobile Communication for Development (M4D) in Africa, *Journal of Information Systems and Technology Management*, 14(3), 431–438.
- Ávalos, C. Pérez-Escoda, A. & Monge, L. (2019). Lean Startup as a Learning Method for Developing Digital and Research Competencies, *Journal of New Approaches in Educational Research*, 8(2), 227-242.
- ZH, M. R. (2001). Technological leapfrogging by developing countries, *EOLSS*.
- Brenner, W. Uebernickel, F. & Abrell, T. (2016). Design Thinking as Mindset, Process, and Toolbox Experiences from Research and Teaching at the University of St. Gallen, in Brenner, W. & Uebernickel, F. (eds.) *Design Thinking for Innovation*, Springer.
- Chang, T. (2004). Supporting Design Learning with Design Puzzles, in Van Leeuwen, J.P. & H.J.P. Timmermans (eds.), *Kluwer Academic Publishers*, 293-307.
- Cisse, M. (2018). Look to Africa to advance artificial intelligence, *Nature*, 562, 461.
- Cuckier D. & Kon F. (2015). Early stage software startup patterns strategies to building high-tech companies from scratch, 22<sup>nd</sup> conference on pattern languages for program design.

- Harwood, W. S. Reiff, R. R. & Phillipson, R. (2005). Putting the Puzzle Together: Scientists' Metaphors for Scientific Inquiry, *SPRING*, 14(1), 25-30.
- Henderson, R. M. & Clark, K. B. (1990). Architectural Innovation: The Reconfiguration of Existing Product Technology and Failure of Established Firms, *Administrative Science Quarterly*, 35, 9-30.
- Hevner, A. R. March, S. T. Park, J. & Ram, S. (2004). Design Science in Information Systems Research, *MIS Quarterly*, 28(1), 75-105.
- Hey, J. Linsey, J. Agogino, A. & Wood, A. (2008). Analogies in Creative Design, *International Journal of Engineering education*, 24(2), 283-302.
- Hirschfield, R. Steinert, B. & Lincke, J. (2011). Agile software Development in Virtual Collaboration Environments, in *Design thinking: Understand improve apply*, Meinel C. and Leifer L. Eds., Springer, 197-218.
- Iyamu, T. (2021). The Role of African Proverbs in Advancing Information Systems Research, *The African Journal of Information Systems*, 13(4), 476-492.
- Kerry, M. B. & Myers, B. A. (2017) Exploring Exploratory programming, *IEEE symposium on Visual Languages and Human Centric Computing*.
- Local researchers & Easton. P. (2004). Participatory Management and Local Culture: Proverbs and Paradigms, in *Indigenous Knowledge: Local Pathways to Global Development*, Knowledge and Learning Group, World Bank, 128-131.
- Mambo, W. N. (2017). Adapting Information Technology Innovations to Country Context, *IJCIT*, 6(1), 27-31.
- Mambo, W. N. (2020). Designing Per-Poor System of Innovation Proverbs, *Inkanyiso, Jnl Hum & Soc Sci* 12(2), 212-229.
- Mambo, W. N. (2021). Innovatively Exploring Proverbs with Transdisciplinarity, *African Journal of Applied Research*, 7(2), 137-154.
- Mambo, W. N. (2022). Aligning Software Engineering and Artificial Intelligence with Transdisciplinary, *Transdisciplinary Journal of Engineering & Science*, 21-34.
- Miah, S. J. & Genemo, H. (2016). A Design Science Research Methodology for Expert Systems Development, *Australasian Journal of Information Systems*, 20.
- Nayebare, M. (2019). Artificial Intelligence Policies in Africa: Over the Next Five Years, *XRDS*, 26 (2), ACM.
- Parjanen, S. (2008). Distance and Proximity as Source of Innovation in Regional Innovation Processes, *Conference on Regional Development and Innovation Processes*, Porvoo, Finland.
- Perkusich, M. e Silva, L. Costa, A. Ramos, F. Saraiva, R. Freire, A. Dilorenzo, E. Dantas, E. Santos, D. Gorgonio, K. Almeida, H. & Perkusich, A. (2019). Intelligent Software Engineering in the Context of Agile Software Development: a Systematic Literature Review, *Information and Software Technology*.
- Pries-Heje, J. & Baskerville, R. (2008). The Design Theory Nexus, *MIS Quarterly*, 32(4), 731-755.
- Ries, E. (2011). *The Lean Start-up: How Today's Entrepreneurs use Continuous Innovation to Create Radically Successful Businesses*. Crown Books.
- Riesco, A. M. Artaso, M. A. Belandia, M. & Pickin, S. (2014). Still in the First Steps of a Barefoot Knowledge Engineering, *IEEE*.
- Strusani, D. & Hounbonon, G. V. (2019). The Role of Artificial Intelligence in Supporting Development in Emerging Markets, *International finance corporation*.

- Vaishavi, V. & Kuecher, W. (2009). *Design Research in Information Systems*, Association of Information Systems.
- Vaishavi, V. & Kuecher, W. (2015). *Design Science Research Methods and Patterns Innovating Information and Communication Technology* 2<sup>nd</sup> Edition, Taylor & Francis Group.
- Wieringa, R. J. (2014). *Design Science Methodology in Information System and Software Engineering*, Springer Verlag.
- World Bank, (2004). *Indigenous Knowledge: Local Pathways to Global Development*, Knowledge and Learning Group, World Bank.
- Zhuge, H. Ma, J. & Shi, X. (1997). Abstraction and Analogy cognitive space: A software process model, *Information and software technology*, 3.

**Conflict of interest:** Author declares there is no conflict of interest.

**Editor-in-chief:** Edson Luiz Riccio

**Data Availability Statement:** All data generated or analysed during this study are included in this published article.