Evolutionary Computing Techniques and Methods

Naresh E[1*], Shiva Darshan S L[2*], Srinidhi N N[3*], Ananda Babu J[4]

[1*]Department of Information Technology, Manipal Institute of Technology Bengaluru,Manipal Academy of Higher Education, Manipal, India
[2*]Department of Information Technology, Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, India
[4]Department of Information Science and Engineering, Malnad College of Engineering, Hassan, Karnataka
[3*]Department of Computer Science and Engineering, Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, India.

Abstract
We have come to realize that as our needs evolve, software needs to evolve to better develop, understand and move forward in technological history. The need for everchanging solutions for our everchanging needs gave rise to a software engineering model that has been used since the 1950s. There are two main classifications of Evolutionary Techniques in Software Engineering. Evolutionary computing can be defined as a system that can adjust to evolving requirements and the world in which it operates continuously. This can be interpreted as a metaphor for Darwin's theory. but this concept has been used fervently since the 1950s to explore, improve and gain better insight into technological sciences and algorithms. Evolutionary development can be vaguely described as a mixture of iterative and gradual models in the software development lifecycle. This model splits down the production cycle into simpler, gradual cascade models, with consumers gaining access to the product after each cycle. Users provide input to help develop the product, resulting in quantifiable deliverables that can be used to promptly check and confirm their requirements. In this article, we will be exploring these various methods that have been studied throughout the years and their real-world applications.

Keywords: Evolutionary Computing, Evolutionary Development, Genetic algorithm, Traditional Evolutionary Programming, Genetic Programming, Evolutionary Prototyping, Quantifiable, Throw-Away prototyping, Spiral Model Evolutionary Method, Agile Model, SBSD - Search-Based Software Development.

1. Introduction

Today's Software landscape is marked by continuous transition, incredibly short deadlines, and a heavy focus on customer–user satisfaction. The Evolutionary Model can be seen as one of the best solutions to combat all three of these factors and deliver the best results. This motivated me to select this topic to research and also to present to the class how important this model can be in software engineering.

Evolutionary computation is the analysis of algorithms that use supervised selection to look for a set of solutions to a particular problem.

Evolutionary Development is a technique that was designed to better improve on various primitive development models. It is in a way the form of an Agile development model as it incorporates customers feedback into the lifecycle. It is also a combination of iterative and incremental development models.

2. Related Work

This is detailed research on the historical timeline of software development techniques and how evolutionary computation [6] was introduced to software engineering. It explains how Alan Turing first introduced the idea that evolutionary concepts from Darwin could hold a sympathetic counterpart in computation and how John Holland's subsequent experiments on evolutionary computing popularized the field.

A deep analysis of the two computing techniques in evolutionary computing. The primitive evolutionary computing algorithm [7] and genetic algorithm [7] are explored and their applications are discussed. Traditional computing keeps the program intact while correcting the parameters. Genetic algorithms apply a learning technique to modify programs to optimize the process model better [8].

Various approaches have been used to solve feature selection [1][2] problems, with evolutionary computation techniques gaining a lot of attention recently and showing some improvement. However, there are no detailed recommendations on the advantages and disadvantages of alternative methods. We explore other methods such as SVMs, ACO graph methods, GAs, etc can be used and combined to get better and optimized solutions for various problems.

We examine problems in information retrieval [9][10][11] that have been solved with evolutionary algorithms. Some of the latest approaches are listed in detail for some of these issues. A broad range of problems has been solved using *Genetic algorithms*, including automatic document indexing, clustering, query definition, image retrieval, etc. Information fusion, text analysis, multimedia processing, ranking, and web mining will all benefit from *Evolutionary algorithms*.

The importance of Evolutionary methods in scheduling and combinatorial optimization [3] faces a few key challenges. Practical implementations and contexts in which solutions are implemented are investigated to support evolutionary computing methods to understand how EC methods can perform in operation [12]. By Merging [4]AI, ML, and other concepts we can deal with critical problems such as cross-referencing, multiple objectives, and dynamic changes. [5] Hyper-Heuristics like genetic programming algorithms have good potential to provide an optimized and scalable solution.

3. Evolutionary Computing

Evolutionary computing is a process model that consists of a family of global optimization algorithms inspired by the Darwinian Theory of evolution for a biological Ecosystem and its implementation.

Software scientists widely explore two evolutionary strategies:

Traditional Evolutionary Programming is interested in infinite-state automata adaptation for machine learning tasks. This application requires specialized representation and operators. On the other hand, contemporary evolutionary programming uses a stochastic variant of the plus replacement strategy and can use any representation and different evolution engines.

Genetic Programming In machine learning and modeling activities has a specific application field. Parse-trees of formal logical expressions representing a model or process are a natural representation. The crossover and mutation operators have been modified to function on trees (with varying sizes). Genetic Algorithms are "inherited" by the evolution engine (For a long time, Genetic Programming was thought to be Genetic Algorithms with a representation of trees.). The branch of genetic programming that deals with the programmed development of programs is known as genetic programming.

3.1. Applications of Evolutionary Computing

- To create highly efficient process models which consist of fault tolerance, error correction, and controlled learning rate.
- In the medical industry, these methods can be applied to create software to recognize dangerous diseases and predict if a person is affected or close to being affected by a malady.
- It can be used in mathematics to equate and predict tough to solve iterations and problems.

3.2. Traditional Evolutionary Programming

- Image processing algorithms used traditional evolutionary computing methods to improve the recognition of features and optimize facial recognition.

- It was used in the Artificial Intelligence generation model where the program's structure is fixed, and the parameters are changed according to the learning process.

3.3. Genetic Programming

- It has significant use in the Machine Learning field where genetic algorithms are used for feature selection, prediction models, and clustering algorithms.

- The music industry has recently started to implement genetic programming to analyze and create a self-learning program to classify and create tunes and also predict user reactions.

- Object detection in image processing algorithms has greatly been influenced and optimized using genetic programming models.

4. Evolutionary Development

According to the Evolutionary Development model, work should be broken down into manageable parts, prioritized, and then distributed to customers one by one. The number of parts is immense, and it is proportional to the number of customer deliveries. The key benefit is that the customer's trust grows because he is continually provided with quantifiable products or services to check and confirm his specifications from the project's outset. All work is broken down into manageable workpieces, and the model allows for changing requirements.

There are mainly two methods that are widely used when it comes to the Evolutionary Development model:

Evolutionary Prototyping is a technique for creating durable prototypes that are continually refined to reflect a design update, potential product, or a state-of-the-art demonstration. They're a costly investment that's popular in large-scale industries like the automotive industry, where goods must be extremely refined before being published.

Throw-Away prototyping is also a similar method that instead focuses on creating a low-cost, fast prototype developed to model an idea or function. This is generally reserved for small-scale productions and ventures in the early stages.

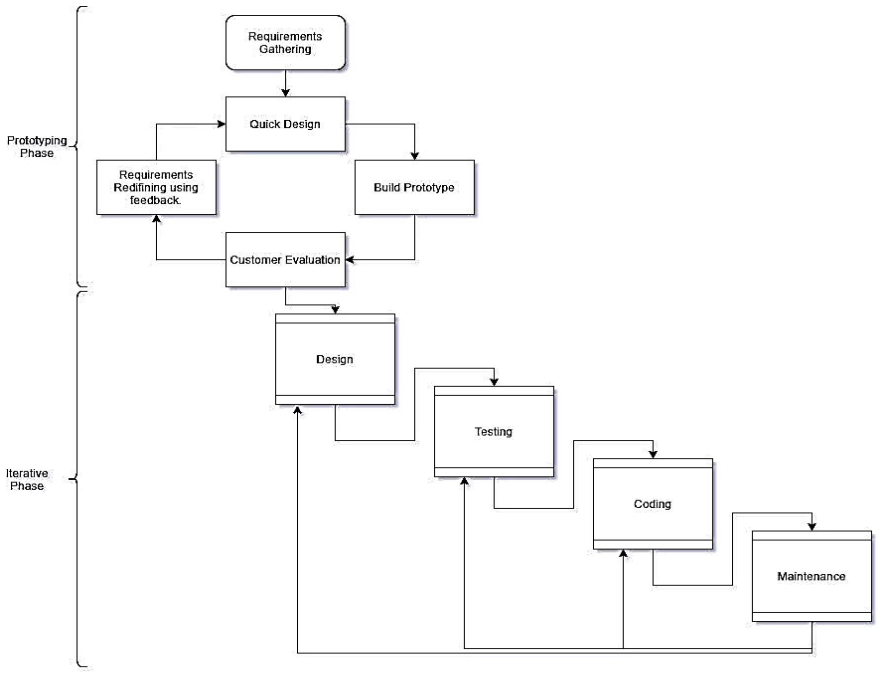

Figure 1: Evolutionary Prototyping

Spiral Model Evolutionary Method is an evolutionary software process model, first proposed by Boehm, that combines the iterative feature of prototyping with the regulated and systematic aspects of the linear sequential model. It implements the capability of rapid production of new software versions.

This process starts with a planning phase and cycles through risk management/analysis, Requirement's analysis and re-evaluation, Development of product/prototype, and Customer feedback/suggestions to come back to the planning phase and thus spirally proceeds to develop a well-processed product with prioritized requirements according to the needs.

The Evolutionary model allows a user to experiment with a partially constructed framework, which eliminates error by extensively testing [13][14] the core modules.

Applications of Evolutionary Model:

- It can be used in big projects where modules for gradual deployment are readily accessible.
- The evolutionary paradigm is widely used if a client needs to start using the key functionality right away rather than waiting for the entire software.
- Since the framework can be conveniently divided into units in terms of elements, the evolutionary paradigm is often used in object-oriented software development.

Prototyping has real-world applications in the automotive field and the Drone industry.

- In the case of the automotive industry, a customer is presented with a design prototype, and based on feedback the product is created.
- In the drone industry, a prototype can be created by taking prioritized requirements and presenting them to the customer and making improvements.

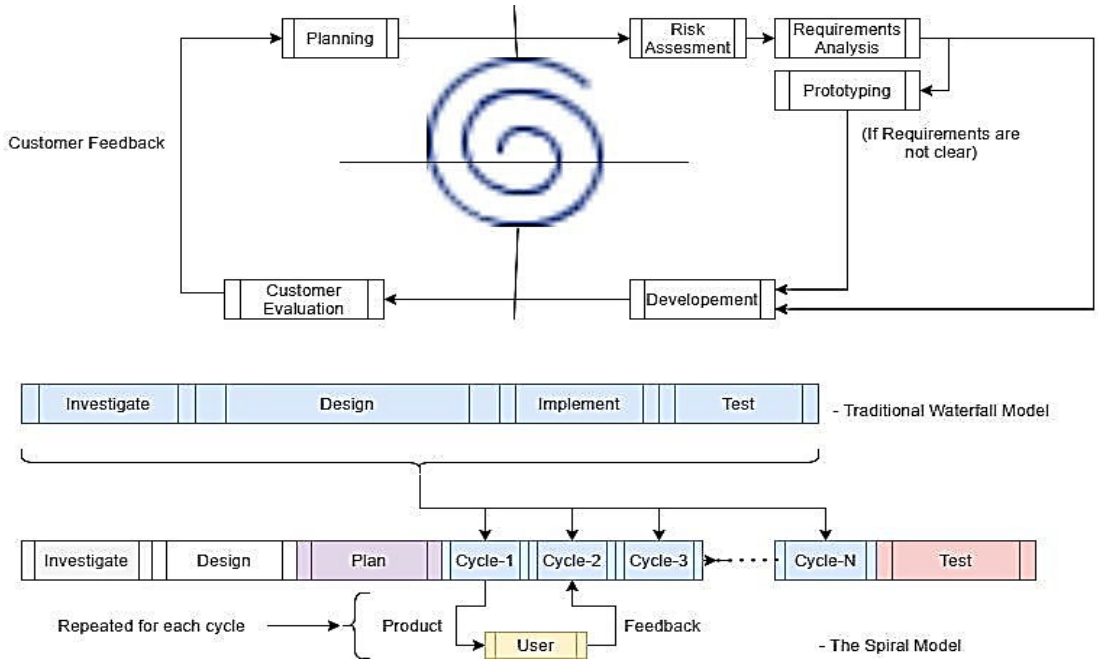

Figure 2: Spiral Model for Evolutionary Development

The **Spiral-Model** has various real-world applications in the software, military, and space industries.

- Software such as Microsoft Office was created with a spiral model and has shown incredible improvements throughout the years to vouch for this particular model for software development.
- The risks associated and cumulative costs for creating missiles and Satellites are a huge factor in why spiral models are used in military and space industries to identify and evaluate risks and prioritize final product requirements.

5. Conclusion

Software development models are crucial in the production of every product. The concept of evolution in Software Engineering is also a significant part of the Software Engineering Process as it is one of the most widely used models in today's world.

The genetic programming concept has brought welcoming change in software by reducing time and space complexities and improving the machine learning process.
The comparison of throwaway/prototyping models shows that both models are used for different requirement-specific projects and the comparison b/w these and The Spiral model results in the same. The Spiral model is preferred for large-scale developments where an agile process is preferred.

References

1. B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A Survey on Evolutionary Computation Approaches to Feature Selection," in IEEE Transactions on Evolutionary Computation, vol. 20, no. 4, pp. 606-626, Aug. 2016,
2. F. Dong-Hua and L. Meng, "Research on Modeling of User Learning Interest Based on Evolutionary Computation," 2018 4th Annual International Conference on Network and Information Systems for Computers (ICNISC), Wuhan, China, 2018, pp. 273-277,
3. X. Luo, Y. Wan and X. He, "A comparison of modified evolutionary computation algorithms with applications to three-dimensional endoscopic camera motion tracking," 2017 IEEE International Conference on Image Processing (ICIP), Beijing, 2017, pp. 3840-3843,
4. H. Takenouchi and M. Tokumaru, "Interactive Evolutionary Computation System Using Multiple Users' Gaze Information Considering User's Partial Evaluation Participation," 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST), Morioka, Japan, 2019, pp. 1-5,
5. S. Nguyen, Y. Mei, H. Ma, A. Chen and M. Zhang, "Evolutionary scheduling and combinatorial optimization: Applications, challenges, and future directions," 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 2016, pp. 3053-3060,
6. Mark Harman, "Software Engineering Meets Evolutionary Computation." Published by the IEEE Computer Society, 2011 IEEE, Oct. 2011, 0018-9162/11
7. A.E. Eiben, M. Schoenauer, "Evolutionary computing." 0020-0190/02/$ – Published by Elsevier Science B.V, 2002, 0020-0190/02
8. Divya Tanwar, "SOFTWARE DEVELOPMENT MODELS AND THEIR REAL ASPECTS" Published by the International Journal of Advance Research in Science and Engineering, Vol. No. 5, Issue No. 03, March 2016, ISSN 2319-8354
9. O. Cordón, E. Herrera-Viedma, C. López-Pujalte, M. Luque, C. Zarco, A review on the application of evolutionary computation to information retrieval, International Journal of Approximate Reasoning, Volume 34, Issues 2–3, 2003, Pages 241-264, ISSN 0888-613X.
10. Cagnoni, Stefano & Poli, Riccardo & Smith, George & Corne, David & Oates, Martin & Hart, Emma & Lanzi, Pier Luca & Willem, E.J. & Li, Yun & Paechter, Ben & Fogarty, T.C.. (2000). Real-World Applications of Evolutionary Computing.
11. T. Back, M. Emmerich and O. M. Shir, "Evolutionary algorithms for real-world applications [Application Notes]," in IEEE Computational Intelligence Magazine, vol. 3, no. 1, pp. 64-67, February 2008,

12. Tzung-Pei Hong, Chuan-Kang Ting, Oliver Kramer, Theory and Applications of Evolutionary Computation, Published in Applied Computational Intelligence and Soft Computing, Hindawi Publishing Corporation 2010, 2010/08/30, 360796, 2010, 1687-9724.
13. Rayudu, Dadi Mohankrishna. "Naresh. E and Dr. Vijaya Kumar B. P, "The Impact of Test-Driven Development on Software Defects and Cost: A Comparative Case Study"." International Journal of Computer Engineering and Technology (IJCET) 5, no. 2 (2014).
14. Naresh.E and Sandeep Kumar Kalaskar "A Novel Testing Methodology to Improve the Quality of Testing a GUI Application", MSRJETR, Vol.1 No.1, Page no 41-46, July 2013.