



A Genetic Based Intelligent Approach to Estimate Software Release Using Agile

Prince Rana

Deptt. of CSE
Lovely Professional University
Jalandhar, Punjab(India)

Naveen Bilandi

Assistant Professor
Deptt. Of CSE, DAVIET
Jalandhar, Punjab(India)

Sumit Mahajan

Deptt. of CSE
DAVIET, Jalandhar
Punjab(India)

Sudip Karmakar

Deptt. of CSE
NIT, Jalandhar, Punjab
(India)

Abstract—Software Release is the final step of software development in which the software is delivered to the client. But when the release will be performed is the main question. The main interest of management and other stakeholders is to estimate the software release on the earlier stages of software development. The proposed work is in same direction to predict the software release. In this present work we are using the Genetics as the prediction as well as optimization approach to get the software release date based on available available resources and their utilization. The present system is an intelligent system that will return the result in the form a detailed PERT chart.

Keywords— PERT, Software Release, Genetics, Resources, Intelligent System

I. INTRODUCTION

Software release management is an important key technology for distributing the project/product to the customer. The key success factor of any software product lies in how delicately the product is released to the customer.

“Release management is the process of planning, building, testing and deploying hardware and software, the version control and storage of software.”[1] Release management process becomes vital in product development & deployment, it basically manages the frequency of product releases and their levels, i.e. full release or patches. Release management is not just what goes into the product development environment but also how something goes into the product development environment. Implementing a true release model results in two businesses benefits, reduction in overall cost and improved customer satisfaction. Intech process automation (www.intechww.com) is in the process of product development. It is required to manage multiple releases of a product. At the moment, internal releases are continually produced. Software configuration management practices are formally followed for development and producing internal releases. No special consideration is being given to handle multiple releases.

Figure 1 shows the basic phenomenon followed to perform the software release. The first release is just after the development process and further releases based on the previous release with required rectification as well as estimation. Each stage of software release increase the optimization process and decreases the user efforts. For each

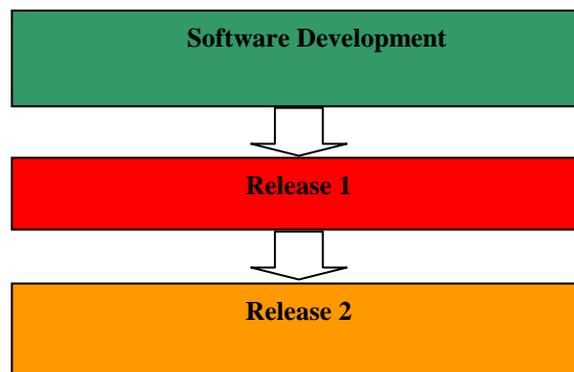


Figure 1 : Software Release

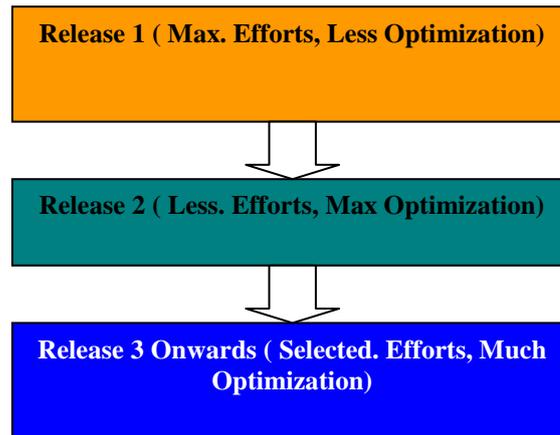


Figure 2 : Software Release

Figure 2 shows the general theory behind the relation between the software releases and relative estimation of efforts and the optimization.

II. GENETICS

The genetic algorithm is a method for solving both constrained and unconstrained optimization problems that is based on natural selection, the process that drives biological evolution. The genetic algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm selects individuals at random from the current population to be parents and uses them to produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution. Genetic algorithm can be applied to solve a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, non differentiable, stochastic, or highly nonlinear.

The genetic algorithm uses three main types of rules at each step to create the next generation from the current population:

A genetic algorithm is modeled on genetics and Darwinian evolution, whereas a neural network is based on models of human cognition. One common application of a genetic algorithm is as a function optimizer. Another common application of a genetic algorithm is to evolve organisms that perform well in a given environment. In either application, the GA is based on the survival-of-the-fittest (natural selection) tenet of Darwinian evolution. A genetic algorithm consists of a population of individuals that reproduce (over many Generations) according to their fitness in an environment. Those individuals that are most fit are most likely to survive, mate, and bear children. Children are created by the stochastic application of genetic operators to the (parent) individuals. Individuals of the population, coupled with the genetic operators, combine to perform an efficient domain-independent search strategy that makes few assumptions about the search space. Each individual in a population is a point in the search space.

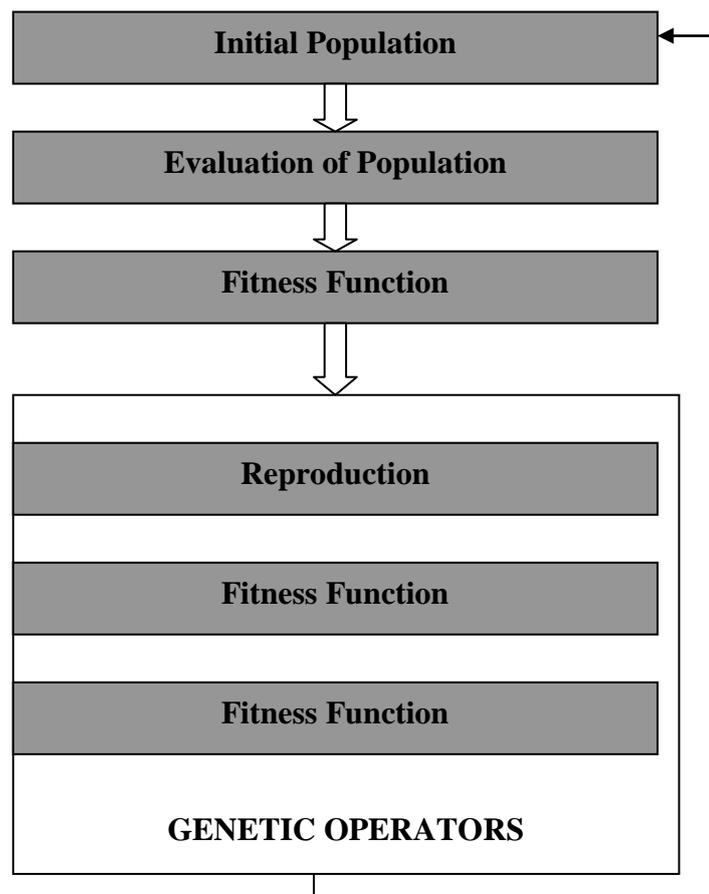
Traditionally, an individual in a GA is represented as a bit string of some length n . Each individual thus represent some point in a space of size 2^n . Given a bit string representation. These representations are all single-stranded in the sense that one piece of genetic material represents an individual. Such representations are termed *haploid*. However, natural genetics makes use of double stranded chromosomes (*diploid*) as well. For example, suppose two bit strings represent an individual:

1010001010
0010101001

These double strands can contain different and possibly conflicting information. In nature, *dominance* is the primary mechanism for conflict resolution. Supposing 1 to dominate 0, the individual *phenotype* can be expressed as:

1010101011

A genetic algorithm will produce offspring using the genetic operators' crossover (recombination) and mutation. Mutation operates at the bit level by randomly flipping bits within the current population. Mutation rates are low, generally around one bit per thousand. Here Figure 1 is showing the basic Architecture of Genetic Algorithm.



GA Architecture

Each stage of Genetics itself is a Research Area. Lot of work is done by many of researchers on each stage of Genetics. Number of problem are being solved and optimized by using this Genetic Algorithm. Here we presenting the comparative analysis of some different operations or algorithms performed on different stages of Genetic Algorithm.

III. LITERATURE SURVEY

In year 2008, Bo Yang, Huajun Hu, and Lixin Ji, Worked on “A Study of Uncertainty in Software Cost and Its Impact on Optimal Software Release Time”. they study the uncertainty in software cost and its impact on optimal software release time in detail. The uncertainty is quantified by the variance of the AC and several risk functions. A risk-control approach to the optimal software release problem is proposed. New formulations of the problem which are extensions of current formulations are developed and solution procedures are established. Several examples are presented. Results reveal that it seems crucial to take into account the uncertainty in software cost in the optimal software release problem; otherwise, unsafe decisions may be reached which could be a false dawn to management^[1]. In year 2004, Annie Ibrahim Rana, Muhammad Waseem Arfi defined a work on “SOFTWARE RELEASE METHODOLOGY: A CASE STUDY”. In this work they analyze different quality parameters related to the release of a product. These parameters should be handled through Software Release Model. The chosen model that supports those parameters is discussed. A controlled environment is tested for those parameters through the use of this model^[2]. In year 2007, Pratik K. Biswas, “Autonomic Software Release Management for Communications Networks”. They introduce r introduces the problem of Software Release Management for communications networks, proposes an agent model and a novel protocol for Autonomic Software Release Management, and discusses a prototype Autonomic Software Release Management system that can be effectively used by the service providers to upgrade and maintain their customers' networks[3].

In year 2009, Xiaohua Wang, Liping Zou has introduce an agile based work on “Decision Model for Agile Software Release”, Analyzing the agile software development process, combining the evolution laws of agile software project, applying AHP method, an integration assessment model for agile software release is presented. The index system for

agile software release plans is built, and its target layer, criterion layer and scheme layer are specified. Comparison matrix and judgment matrix are constructed with three—demarcation analytic hierarchy process (TDAHP). Combining with the views of agile experts about index matrix and weight, the qualitative and quantitative analysis about software release scheme can be carried out, and an optimal scheme can be selected. Finally, a numerical example is given, which shown this model is scientific and feasible[4].

In year 2010, Ahmed Nagy, Mercy Njima and Lusine Mkrtchyan defined work for “A Bayesian Based Method for Agile Software Development Release Planning and Project Health Monitoring”, have designed and developed a project health measurement model to evaluate the factors affecting software development of the project. Author used Bayesian networks (BNs) as an approach that gives such estimation. Author present a quantitative model for project health evaluation that helps decision makers make the right decision early to amend any discrepancy that may hinder on time and high quality software delivery[5].

In year 2010, Ville Heikkila, Kristian Rautiainen proposed a work on “A Revelatory Case Study on Scaling Agile Release Planning”, He conducted a case study to explore how the new release planning method was executed. Author gathered data by observing two release planning events, observing event planning meetings, and by conducting surveys. The events were attended by approximately 140 stakeholders, including over 10 development teams, who spent several days in a common space. The participants liked the method and considered it efficient. This revelatory case study provides the first detailed empirical description of this emerging method for multi-team agile release planning[7]

In year 2011, Tobin J. Lehman and Akhilesh Sharma presented his work on “Software Development as a Service: Agile Experiences”. With many decades of software development experience in Presented group, Author have used many different techniques for conducting software projects. In SDaaS, the style of interaction between the client team and the development team is a natural fit for Agile Software Development methods. Although Presented conditions do not fit the ideal Agile Software Development project profile[6], Author have consistently experienced significant gains from taking the approach of iterative and incremental development. Author have used the Waterfall, or plan-based, method many times for past projects, and Author have consistently paid a significant price for the “Big Design Up Front” problems that go with that type of schedule[8].

IV. PROPOSED WORK

The proposed work is work is about the planning of the software release on the basis of Acquired Information about the project. This information is respective to the Available resources as well as Required Resources. On the basis of this an Agile Method will be appied to automate the proposed work. Authosr are using an intelligent approach to Estimate the Software Release and the result will be presented in the form of gantt chart. The chart will arrange the resource in a sequence.

In this work we augment the scope of release planning by examining details of the resource allocation necessary to actually develop the features. For that, We assume that each feature is decomposed into a sequence of tasks such as design, implementation, and testing. These development tasks can be defined to an even more fine-grained level. In addition, managerial support and/or other tasks can be considered here as well. Given the analysis of the problem's complexity, we have concluded that We need a specialized solution approach. Presented two-phase approach, called Optimized approach, combines the strength of special structure integer linear programming (Phase 1) with the power (Phase 2). The advantage is twofold. First, from Phase 1, we can generate an upper bound for the maximum value achievable. This allows an evaluation of the solution generated in Phase 2. Second, the solution obtained from Phase 1 is used to restrict the set (permutations of N features) to the set used in the GA of Phase 2. This can significantly reduce the computational effort and allows solution of problems of small and medium size. Phase 2 can be applied without application of Phase 1, but the search space would be substantially larger in this case. The restriction provided by Phase 1 is heuristic in its nature

V. CONCLUSION AND FUTURE WORK

In this proposed work we have estimated the schedule of a software release using the Genetic approach. We can extend this work by generating the PERT chart of the same system. We can generate a complete software development plan by using the same approach. The System can be extended for other software development plans for the system with each plan estimation we need to define the certain attributes with respective decision values.

REFERENCES

- [1] M.P. Ware F.G. Wilkie, “The Use of Intra-Release Product Measures in Predicting Release Readiness, 2008 International Conference on Software Testing, Verification, and Validation 0-7695-3127-X/08 © 2008 IEEE Clausen, T., Jacquet, P., Optimized Link State Routing Protocol (OLSR) - RFC 3626, 2003
- [2] Jeff W. Boote, Andreas Hanemann, Loukik Kudarimoti, Panagiotis Louridas, Lu 's MartaSixth, “Quality Assurance in perfSONAR Release Management” International Conference on the Quality of Information and Communications Technology 0-7695-2948-8/07 © 2007 IEEE

- [3] Shinji Inoue and Shigeru Yamada, "Optimal Software Release Policy with Change-Point" 978-1-4244-2630-0/08 ©2008 IEE
- [4] Bo Yang, Huajun Hu, and Lixin Ji, "A Study of Uncertainty in Software Cost and Its Impact on Optimal Software Release Time", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 34, NO. 6, NOVEMBER/DECEMBER 2008 0098-5589/08 2008 IEEE
- [5] Annie Ibrahim Rana, Muhammad Waseem Arfi, "SOFTWARE RELEASE METHODOLOGY: A CASE STUDY", 2004
- [6] Pratik K. Biswas, "Autonomic Software Release Management for Communications Networks", 1-4244-0799-0/07/\$25.00 ©2007 IEEE
- [7] Xiaohua Wang, Liping Zou "A Decision Model for Agile Software Release", 978-1-61284-666-8/11 2011 IEEE
- [8] Ahmed Nagy, Mercy Njima and Lusine Mkrtyan, "A Bayesian Based Method for Agile Software Development Release Planning and Project Health Monitoring", 2010 International Conference on Intelligent Networking and Collaborative Systems 978-0-7695-4278-2/10 © 2010 IEE
- [9] Tobin J. Lehman and Akhilesh Sharma, "Software Development as a Service: Agile Experiences", 2011 Annual SRII Global Conference, 978-0-7695-4371-0/11 © 2011 IEE
- [10] Ville Heikkila, Kristian Rautiainen, "A Revelatory Case Study on Scaling Agile Release Planning", 2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications, 978-0-7695-4170-9/10 © 2010 IEEE
- [11] Shi Zhon, "Agile Planning and Development Methods", 978-1-61284-840-2/11 ©2011 IEEE
- [12] Xiaohua Wang, Liping Zou, "A Decision Model for Agile Software Release" 978-1-61284-666-8/11 2011 IEE
- [13] Muzaffar Iqbal and Muhammad Rizwan, "APPLICATION OF 80/20 RULE IN SOFTWARE ENGINEERING WATERFALL", 978-1-4244-4609-4/09©2009 IEE
- [14] P. Wongthongtham, N. Kasisopha, E. Chang, T. Dillon, "A Software Engineering Ontology as Software Engineering Knowledge Representation" Third 2008 International Conference on Convergence and Hybrid Information Technology 978-0-7695-3407-7/08 © 2008 IEE
- [15] S. Acun ̄a, N. Juristo, and A.M. Moreno, "Emphasizing Human Capabilities in Software Development," IEEE Software, vol. 23, no. 2, pp. 94-101, Mar./Apr. 2006.
- [16] M. Crissis, M. Konrad, and S. Shrum, CMMI—Guidelines for Process Integration and Product Improvement. Addison-Wesley, 2006.

BIOGRAPHIE



Naveen Bilandi completed his M.Tech. in computer science from NIT Jalandhar, Punjab(India). He is employed as Assistant Professor in Department of Computer Science & Engineering, DAVIET College, Jalandhar, India. He has a vast experience in the field of Networking, DBMS, Network Security, Cryptography. His major fields are Mobile ad-hoc networks and providing QOS-aware service in MANET. He has teaching and research experience of more than 2 years. He is professional Member of various National and International Associations. He has more than 7 research papers to his credit in various International/National Journals and Conferences.



Prince Rana completed his M.Tech. in computer science from Lovely Professional University Jalandhar, Punjab(India). He is employed as Assistant Professor in Department of Computer Science & Engineering, BCET College, Ludhiana, India. He has a vast experience in the field of Software Engineering, DBMS.



Sumit Mahajan is employee in Department of CSE, DAVIET, Jalandhar, India. He has done B.Tech. in computer science from Thapar University, Patiala, India His major fields are Mobile ad-hoc networks and providing QOS-aware service in MANET. He is professional Member of various National and International Associations. He is actively involved in presenting the papers in national seminars. He has more than 2 research papers to his credit in various International/National Journals and Conferences.



Sudip Karmakar is pursuing B.Tech. from NIT Jalandhar, Punjab, India in computer science and engineering. His research interest include Microprocessor, Networking and Data Handling Systems, Digital Electronics. Ad-hoc networks and providing QOS-aware service in MANET.