

QA process and the moment of outsourcing

Author Tanya Mitkova for Eniak Testing

Before reaching the final users, software products are created by using standard methodology and following specific stages. We would like to introduce and compare shortly two of them - Waterfall and Evolutionary. You will find descriptions of the various software development stages - Project, Alpha and Beta Software, Final Software and Release, and the role programmers and testers play in them. Furthermore, our objective is to show how part of the software testing can be accomplished by a separate independent organization. We have done our best to acquaint you with our views, experience and the resources for testing that we have at Eniak.

Software Development Methods

Before starting a new software development project, its Project Manager usually defines all stages of the project, the resources needed for each of them and the requirements they have to meet. Job planning also depends on the development method - whether the traditional Waterfall Method or the alternative Evolutionary method has been preferred.

The basic characteristics of these two approaches are as follows:

Waterfall Method - This is the classical software development method. It includes the consequence of Task Analysis, Design, Coding, Testing and Release. In essence, this approach relies on each phase of the project being completed to satisfaction before progressing on to the next phase of development. That means the Design starts after the Analysis is over, the Coding comes after all the Functional Specifications are produced and at the beginning of the Testing phase, the team has at their disposal the complete project documentation. The Waterfall Method is appropriate in an environment of custom software development contracts. It allows the client to define in advance the software functionality that he expects. In this case, it is also easier to define the overall price of the project. This approach does not allow too many changes after the customer and the software developer had agreed on the requirements and the specifications. The possibility of planning resources, time and costs at the outset of the project is one of the advantages of the Waterfall Method.

The associated risks come from the fact that the system functionality has to be decided at an early phase of the development cycle. There are only ideas and prototypes about the future product and it is possible to change the ideas in the process of development. Some modules can prove hard for coding and testing and the competitive products are not sufficiently known.

Evolutionary Method - In this method, incremental features are added to the core product. The core needs to be designed in such a way as to ensure smoothness in the addition of new features and stability, inasmuch as the core contains the basic functionality of the system. This is the first module that programmers develop and each next phase adds more functions to it. The core is being tested and debugged fully as a final product, until it becomes stable enough. Then a new functional group is added, followed by another testing cycle. Furthermore, the software development includes consecutive phases of functional enrichment and testing. This continues to the moment of releasing the initial version, which can be presented to the users.

The advantages of this approach include the ability to add new functions, or change those that proved ineffective in previous versions. The product development passes through several test-phases. This eases the determining of the total testing time and fixing the deadline for project completion. With Waterfall, there is a kind of uncertainty due to "testing at the last moment". No one can foresee how many bugs the product contains, how long it would take these to be found, and what time would be necessary to fix them. The projects developed through Evolutionary do

not have such a shortcoming, as all these important parameters can be foreseen much earlier, due to the numerous testing cycles.

The associated risks relate to the adding of a new code to the core, before the testing of the previous version had finished, or before all the old bugs had been eliminated. Projects of this type need testing at a very early stage, combined with a complete testing cycle of the core. Sometimes the Test-leader does not have enough people and lots of the core defects remain. This goes in contradiction with the main principle of the approach, all the new functions to be added to a previously developed stable core, and this may lead to various problems at the final stage.

Short Comparison between the Two Methods

Evolutionary is usually defined as the more contemporary method for project management and software development. Its greatest merits are flexibility, user feedback and ease in the adding of new functions. This approach is implemented mostly in projects, which involve unlimited number of users - the company develops a software product and expects 5,000 to 20,000 independent customers. The efficiency/profit grows with the number of the customers. The unknown end-users are the ones that pay for the software and, if we compare with the Waterfall method, we can say these are the buyers of the new product. It is not possible to know their needs and wishes at the initial development stage. There is a constant need of new versions, as the requirements change with time. The new requirements, resulting in development of new specifications, come from users feedback. Evolutionary is implemented by some of the biggest IT companies, for example, IBM, HP and Microsoft.

The choice of development method depends on the use and nature of the software product. Waterfall is more appropriate for creating industrial and productions software systems. None would buy a vehicle, in which the auto-electronics were developed through the Evolutionary Method, nor would users fly with an airline, which uses a standard other than CMM 5. Reliability, in addition to the need to predict all possible situations before the product reaches its final users, are some of the major requirements in such projects and Waterfall is the one that can assure it.

Our Approach

Our preparatory work, approach and expectations are different when implementing the two different methods-Waterfall and Evolutionary. With Waterfall, we are prepared for a continuous test tour, concentrated mainly on the Alpha and Beta phases. With Evolutionary, we apply a great number of shorter test phases, lots of them Regression Tests. We closely inspect the relation between the newly added function and the core in order to determine the problems faster and more efficiently. We can invite a group of Beta-testers and use their approach and attitude to improve testing efficiency and get a feedback at an early stage of the product development. When forming the testing team, we will try to involve specialists from the sphere, in which the software will be used. For example, if your product is supposed to be used by lawyers, we will do our best to invite a tester, having knowledge and experience in legal matters. We won't expect him to be the best or the leading tester, but to give us guidelines to better understand the specific features of the industry and your future customers.

Software Development Stages

All Software Projects, no matter what development method is used, pass through the following stages:

- a. Design Phase
- b. Alpha Software
- c. Beta Software

- d. Final Product
- e. Release

It is possible to describe the stages using numerous sub-phases. We will assume this definition, as it is the most common one.

Design Phase

This phase involves determination of the requirements, the specifications and the prototypes. Coding starts at the end of the phase. Testing here means verification of the documents, and not of the program codes. The Test Team Manager prepares and analyses various data, defines the necessary hardware devices and software testing tools and may start to make contacts with the Beta-Testers.

How Can We Help You as an Independent Testing Team

At this stage we can offer you a review of the completeness of the User Interface. We can also help you in analyzing the data and choosing the types of testing. In this way we would be prepared for the next testing phases and can plan the resources and testers in accordance with the needs of your project.

b. Alpha Software

This is the most important milestone in software development. The Alpha stage has lots of definitions. The most common assumes that the product is functional, but it is possible some of its options to be incomplete or unstable. The program clearly demonstrates its nature and style. This is the stage of the utmost testing, including a strict consequence of detailed unit, integration and system tests. After registering all the issues, programmers fix them and a new testing phase starts in order to validate that the corrected bugs have not caused problems in other program modules (Regression Testing).

How Can We Help You as Independent Testing Team at the Alpha Stage

This is the most important phase of the software development cycle. Depending on your needs, we can assist you on different levels. We provide services in two ways:

- **As a Basic Testing Team**
- **As Additional Testers**

When acting as a **Basic Testing Team** we would be responsible for the development of the test strategy, test plans and test cases. The most favorable time to start collaboration is the stage of Software Design, or before the Alpha phase.

Test Plan

Making the Test Plan is one of the most important phases in the process of testing. We describe the functions to be tested, the approach, assumptions, evaluation criteria and the necessary resources.

The testing approach has some peculiarities, depending on the software development method. If you have implemented Waterfall, we will build the testing strategy on the basis of your project documentation and specifications. Usually they are completed and detailed and we will perform all necessary tests to assure that the final product meets exactly its initial requirements. If you have applied Evolutionary, or another agile method, we will develop our testing strategy after

analyzing your specification, feature list, user guide. We will also consider the business rules specific for your branch, talk to the customers' authorities and the actual users. Our approach includes parallel work on testing and on the test plan creation. When we work on the test plan we spend an hour or two to try our ideas on the computer. In this way we explore the software, test its abilities, find some of its weak points and save time. Such an approach helps a lot in discovering the most possible problem areas and in defining the system modules that need more profound testing. On this basis we can easily develop and apply the most appropriate test set. On the other hand, we act as the actual users of the product, which brings a valuable feedback. The Evolutionary approach implies earlier start of Beta-testing and close examination of testers' behavior and attitude towards the product.

If you choose our team to be **Additional Testers** in your project, we would follow the Test Plan you have worked out, or consider your instructions and testing requirements. The best time for us to join testing is in the Alpha phase.

Test Lab Equipment

We would configure the devices, which were described earlier in the Test Plan. These can be the devices from our Lab, specific user equipment or, if necessary, additional devices. By establishing an appropriate Bug Reporting System, different types of Forms and Reports, we aim at acquiring better "visibility" of all found, fixed or future faults.

Test Execution

All preliminary developed and verified tests are executed in our Lab. When the registered results differ from the expected ones, we prepare a Problem Report and input it to the Bug Reporting System. In case this is an online system, you can get almost immediate information about the problems. The other possibility is to receive our Reports by e-mail.

Bug Reporting

Bug Reporting is a very important part of the testing. The registered issues have to be described simply and understandably in written. The issues need to be reproducible by other members of the team. Our Reports include the following information: product version, data, problem type and severity, name of the problem, description of its reproduction algorithm, name of the tester, additional information. We emphasize on description of the problem and its reproduction algorithm. You'll get an explanation of the reasons we think it is a problem and how it has been located. As the programmers and the testers are usually working in different offices and this makes it difficult to answer the questions of the programmers, we use the following approach to ensure that the reports are clear. Each report is carefully re-tested by an independent tester, before inputting it in the official Problem Report System. If necessary we develop a new testing case that would reveal the problem more clearly. We have built our Problem Report Methodology on the basis of experience and participation in different projects.

c. Beta Software

The Beta phase assumes that the software can be presented to its final users for evaluation, which does not always mean the project is entirely complete. It might contain some known problems that will be fixed later. The Beta phase definition differs depending on the chosen development method. When implementing Waterfall, **Beta stage** assumes completed coding and testing phases, there are no fatal errors and the User Documentation is 50% completed. No major new tasks are expected for the programmers.

With Evolutionary method, the product enters **Beta** with a complete core. All the properties, ensuring minimal product functionality are coded, but adding new ones might be expected. Beta stage is reached very early, when implementing the Evolutionary Method.

Both development methods involve Beta-testers - independent end-users, which are supposed to use the product and tell you about their experience with it. There are lots of articles discussing Beta-testers and how their attitude might be used. This theme is of particular interest to our team as well.

How Can Eniak Help You as an Independent Organization at Beta Phase

· From Alpha to Beta phase

In case our collaboration has started at Alpha phase, during Beta we retest fixed bugs quickly. It will be good if we complete a round of testing of all devices under all configurations. We analyze software and, on the basis of the already fixed defects, will try to find other unstable parts. At this stage we already are familiar with the software and know how it is expected to function. This is the stage we act as end-users. For instance, if we are testing software for online ordering of computer components, we register, choose products, change the order, confirm the changes and expect the system to respond adequately to all our requests. This testing approach has been described in "Testing Computer Software"¹. It is totally different from the functional tests applied during the preceding phases and helps locating of a considerable amount new bugs. This is a stage we especially like, as it gives us the liberty to work with the product and involve all our imagination for inventing various extraordinary testing scenarios.

d. Final product

After completion of the Beta phase, the product enters the stage of final evaluation and review, before it is ready for the presentation in public. Presumably, all errors have been eliminated, or some of them have been postponed for debugging in the next version. Reliability evaluation is of great importance. It is well known that there does not exist entirely debugged software and some parts of it could have been done better. The reliability evaluation is a complex management decision that should find a balance between the price and the risk. Testers' part here is to reveal what risks might be expected during the next few months, and in this way help the managers to make the right decision.

Final Test

It is executed after the end of debugging and Beta testing, when there are not expected changes to the Interface. The Test Team executes a final series of testing scenarios, acting as final users. It is common an independent organization to service this phase in order to additionally guarantee the functional stability of the product, before the final decision on its completion level.

¹Cem Kaner, Jack Falk, Hung Quoc Nguyen, 1999