

## Netspective Software Development Process

The process is a tailored evolutionary prototyping-based process with hints of agile development. Evolutionary prototyping is a form of software system creation in which developers gather the most complete, best-understood requirements possible; design and implement a prototype; and have the customer evaluate the prototype. The developers glean new requirements and clarify existing requirements based on this interaction. This process is repeated until a working system emerges that encompasses the true set of customer and system requirements.

The process allows for four development phases. During the first phase, some initial work is performed to establish the business case, plan the project, and to better understand the customer needs and wants, as well as to establish a basic understanding of what is required of the new software. The second phase verifies that the team has understood the general direction to which the system needs to aspire. This phase involves some throw-away user interface prototypes created for the purpose of demonstrating to the customer the intentions of the group and the first working prototype. The customer evaluates this prototype and adds and modifies requirements. In the third phase, the first true embodiment of the requirements results in a working application. Finally, during the last development phase, the product is evolved with the new and modified requirements. The system is tested and delivered to the customer.

Specific areas must be addressed throughout the evolution of the software. One of the first items created, when appropriate or needed, is the Concept of Operations. Also in the initial development phase, the project plan is written. Both plans must be maintained as the

development progresses. The Concept of Operations helps to describe the project in the context of the application's users and must be modified to meet the true state of the project and user community. In addition, as schedules, personnel, roles, responsibilities, project descriptions, or related documents are changed, created, added, or deleted during the course of the project, the project plan must be updated to reflect these modifications. The provided Concept of Operations and project plan templates can serve as guides as to what items must be included. The project manager with the assistance of the technical writer should be responsible for the maintenance of the project plan, while the business manager with the assistance of the technical writer is responsible for updating the Concept of Operations.

Privacy and security are aspects must be established from the beginning of the project. They are maintained using the Certification & Accreditation (C&A) template – and to be effective, these policies must not only be documented, but they must also be embodied in the system that is produced.

In addition, configuration management must be performed throughout the project's lifetime. Configuration management is the responsibility of the technical writer and is validated by the project manager. It should be completed as set forth in the configuration management plan, which in turn is based, at least in part, on the provided configuration management template.

Finally, throughout the process, quality assurance is an important area on which to focus. All documents should be checked for errors and go through the process of being validated and verified. In addition, the prototypes themselves must be determined to be free of error and defects. The creation of tests and review of both documentation and code should be done continuously throughout system creation. The Quality Assurance Group leads this effort

## Inception Phase

During the first phase, initial customer input serves as the basis from which the business manager and the technical writer develop the [optional] Concept of Operations using the Concept of Operations template. Using this information, the project manager and technical writer establish the project plan with assistance from the entire group in accordance with the provided project plan template. This allows for the creation and understanding of roles, responsibilities, schedules, and a project description. At approximately the same time, work should be performed to create the security and privacy policies that will guide the project (Certification & Accreditation – C&A). The team collaborates on the creation of the policies, and the policies and the adherence to these policies will serve as a measure of the quality of the system that is produced. This underscores the importance of security and privacy aspects of an eCommerce system.

After the policies are established, an initial set of requirements can be gathered from the customer, users, other software systems, and project team so as to contribute understanding to what the system must do. The requirements gathering effort is led by the requirements engineer under the direction of the project manager, who in turn delegates requirements gathering to the team. The requirements should be documented as set forth in the requirements document template with the technical writer leading the documentation creation effort. The phase ends with a group review / risk meeting to analyze the group's efforts, directions, problems, and solutions. This meeting is performed as given in the risk / review meeting template.

### Inception Phase Accomplishments

- Concept of Operations has been created using the Concept of Operations Template
- Project Plan has been created using the Project Plan Template
- Privacy and Security Policies have been created using the Certification & Accreditation (C&A) template
- Initial Requirements Set has been documented using the Requirements Document Template
- Risk / Review Meeting has been held according to the Review and Risk Meeting Template. Security risks are incorporated into the Certification & Accreditation (C&A) document; business risks may be included in the Concept of Operations document.
- Risks have been identified and corresponding mitigation strategies have been enacted as discussed in the Risk /Review Meeting

### Inception Phase Tools

- MS Office (documentation)
- Adobe Acrobat (generating PDFs of documents for configuration control)
- RequisitePro (requirements management, *optional*)
- Git (revision control of documentation)
- ActiveCollab/Jira (tracking issues, tasks, features) and Wiki for project collaboration

## Elaboration Phase

To aid in understanding and to provide a more concrete communication path between the customer and the development team, the second phase focuses on creating throwaway-user interface prototypes to ensure that development is proceeding in the right direction.

The Software Development Group creates basic user interface screens with input and help from the entire team. These screens are archived by the technical writer according to the appropriate configuration management policies. In addition, and in parallel, the chief architect begins to work toward creating a system design based on the identified requirements.

The user interface screens are taken to the customer to bridge communication between the stakeholders. These screens serve as a tool to allow stakeholders to discuss the system and discover communication breakdowns and missing requirements. This process phase also helps to ensure future development will follow the correct path.

As the customer and developer(s) discuss the screens and the system, new or modified requirements may be discovered. Such requirements are documented in full by the technical writer as a complementary section to the original requirements document. The phase ends with a risk / review meeting to provide a common understanding of the status of the project, as well as uncover and rank risks, create risk mitigation strategies, and judge the effectiveness of risk mitigation strategies. The system screens are archived and not used again.

### Elaboration Phase Accomplishments

- User Interface Prototyping has been performed
- Initial System Design has begun using the Design Document Template
- Customer has evaluated the prototypes of the interface
- Requirement set has been modified according to the Requirements Document Template with the feedback from the interface evaluation
- Risk / Review Meeting has been held according to the Review and Risk Meeting Template
- Risks have been identified and corresponding mitigation strategies have been enacted as discussed in the Risk /Review Meeting

### Elaboration Phase Tools

- MS Office (documentation, generating paper prototypes)
- Git (revision control of prototype, documentation)
- ActiveCollab/Jira (tracking issues, tasks, features) and Wiki for project collaboration
- IDEs will vary depending on the development language chosen

## Development Phase, First Major Iteration

In accordance with the new requirements and customer's comments, the system design is modified and completed as the bridge between the existing set of requirements and prototype implementation. The chief architect is responsible for overseeing the design completion. As the system design takes form, the lead programmer delegates the implementation work that arises from the system design and together, the team codes. The quality assurance expert creates tests for the software components and system based off of the requirements. The system programming, debugging, and testing tasks are performed until the requirements are met and the design is fulfilled.

When the prototype is completed, it is shown to the customer. The customer looks over the created system and along with the developers, may uncover requirements that are new or requirements that need to be changed. As before, the new or modified requirements are documented in full by the technical writer as a complementary section to the original requirements document. The phase ends with a risk / review meeting to provide a common understanding of status of the project, as well as uncover and rank risks, create risk mitigation strategies, and judge the effectiveness of risk mitigation strategies.

### First Iteration of Development Phase Accomplishments

- The system design is completed according to the Design Document Template
- The first prototype that embodies the requirements and design has been created
- The first prototype has been shown to and evaluated by the customer
- New or modified requirements are identified and documented as provided for by the Requirements Document Template
- Risk / Review Meeting has been held according to the Review and Risk Meeting Template
- Risks have been identified and corresponding mitigation strategies have been enacted as discussed in the Risk /Review Meeting

### Development Phase Tools

- Git (revision control of prototype, documentation)
- ActiveCollab/Jira (tracking issues, tasks, features) and Wiki for project collaboration
- IDEs will vary depending on the development language chosen

## Development Phase, Second Major Iteration

In accordance with the new requirements and customer's comments, the system design is modified and completed as the bridge between the existing set of requirements and prototype implementation. The chief architect is responsible for overseeing the design modification. As the system design is reworked, the lead programmer delegates the implementation work that arises from the system redesign and together, the team codes. The quality assurance expert creates tests for the software components and system based off of the new requirements. The system programming and debugging is performed until the requirements are met and the design is fulfilled.

At the conclusion of system implementation, the developers thoroughly test the system, using the tests that have been created by the Software Quality Assurance Group and including any tests to complete testing coverage as necessary. As defects or bugs are found, the system programmer orchestrates the fixes needed to correct problems. After the system is tested, a review / risk meeting is held to wrap up system creation. Unresolved issues are discussed. Finally, the completed system is delivered to the customer. This system should encompass the needed functionality and the approved system qualities as dictated by the customer.

### Second Major Iteration of Development Phase Accomplishments

- The system design has been updated in a form consistent with the Design Document Template
- The final prototype that embodies the requirements and design has been created
- The final prototype has gone through extensive testing as noted in the SQA Template
- Risk / Review Meeting has been held according to the Review and Risk Meeting Template
- Open issues have been identified and discussed
- The final application has been delivered to the customer

## **Areas of Special Concern**

Just as there are reasons why an evolutionary prototyping base was chosen for this model, there are areas where practitioners using this model need to pay careful attention. If these drawbacks can be identified, they can be monitored to ensure the process results in a successful project. Such areas of special concern include:

### **Difficulty of Control**

Software engineers who use evolutionary prototyping often face a process that seems difficult to manage and control. The prototyping literature demonstrates that management of evolutionary prototyping is a major contributor to the building of 'good' software. Control of this process lays in solid management practices, as it can be difficult for all stakeholders to agree on the direction and objectives of the software development. The contents of the next prototype may also be hard to determine as a number of revisions and decisions may need to be made by management. Finally, it is often hard to judge the level of completeness of the prototyped system as prototyped systems often lend themselves to being either over-evolved or delivered prematurely. Various methods abound in handling the difficulty of control issue. Research has been done in the areas of limiting iteration phases and using risk analysis for the purpose of controlling the evolutionary process.

### **Functionality Creep Possibility**

A major benefit of evolutionary prototyping for a process model is that it leads to systems that meet known user requirements and yet can change to meet the user requirements that are discovered later. This helps to produce systems that the customers want and need. However, this too, can become burdensome as customers often try to continually request that new features be added – features that will add to development time and costs. Stricter

time limits and limits on the number of iteration phases are sometimes used to mitigate this drawback.

### **Poorly Structured Applications**

Traditionally, evolutionary prototypes have had a certain amount of poor structure associated with them. This structure deficiency is mainly the result of the iterative process of the evolutionary model. Often, the structures that result from evolved prototypes are considered to be inefficient when compared to their linear system counterparts. Care must be taken when cycling through prototype refinement processes so that the design is kept clean.

### **Lack of Continued Commitment from Users**

End user involvement is a must-have for any evolutionary prototyping based project. End users are an integral part of the process of refining requirements and evaluating prototypes. They must be involved for longer sustained periods than is traditionally required by other processes. Users have to be aware of the status of the prototypes and should know what to expect from the prototypes that they are shown. This is necessary as users sometimes view prototypes as the 'real' systems, even when the systems are not complete or fully specified.