



White paper

It seemed like a lifetime

Application Lifecycle Management

www.stdtime.com
Scoutwest, Inc.

It seemed like a lifetime

Did you know that products have life expectancies, just like other living beings? They are given life at the moment of conception, and grow to maturity and to their final demise at some time in the future. The care and feeding of these organisms is the topic of this chapter. The proper term is Application Lifecycle Management (ALM), and covers the details of managing a product from cradle to grave.

The principles of ALM are integral to mature development disciplines. Experienced development organizations employ these principles to ensure quality in the products they develop.

Application Lifecycle Management:

- Project Management
- Project Tracking
- Requirements Planning
- Design and Development
- Issue Management
- Quality Assurance
- Release Management
- Seasons of Life

Project Management

A certain degree of management overhead will benefit any development effort. Time spent thinking through, and documenting a project, will be regained in the increased efficiency it yields. Project planning involves asking the questions: who, what, and when in great detail. Projects are often broken down into phases such as Requirements gathering, Research, Design, Development, Release. Early phases are strategic, while later phases are more tactical. Planning in early stages benefits later tactical activities. Clearly, the structuring of a development effort has an overhead that takes time away from the project. The theory is to

purposefully insert some structured overhead that will increase overall efficiency, and will ultimately lead to a shorter schedule. Leaving out structure leads to longer schedules because product rework will result from the lack of planning. The trick is to find the proper mix.

Project Tracking

Does your organization really know how long to expect for each project phase? Do you have pressures that demand results in time, quality, or budget? Do you know the optimal mix of time for each phase that guarantees acceptable quality within your budget of time and capital? This knowledge comes from experience and tracking actual metrics. Do you use technical instruments to help you measure these metrics? Project tracking is the microscope of the product development trade. It magnifies the details that make your product profitable. When you track project time and status, you learn the anatomy of your project, and can use this knowledge when estimating future projects.

Requirements Planning

Feature requirements and version releases are delicate issues. The fragile balance of meeting customer needs in each product release is important, and must be managed closely. Releasing products without proper requirements is like attempting to hit a moving target. Your developers will aim happily at each moving target until a critical mass realizes that something is wrong. Then they'll lose faith in management because they'll know that the requirements were not planned properly. During that time, you'll lose time and miss critical market windows. There are simple tools to help in this process. Use them. The results of the requirements phase will become very important later because good planning in upstream phases prevents costly rework in downstream phases.

Design and Development

With a good design, development should be as much like manufacturing as possible. The development phase is greatly simplified because the proper

amount of planning was done. However, humans are not infallible. Even the best designs leave the development phase with a certain number of unknowns. These unknowns must be worked out during the development phase. The act of redesigning during development is known as thrash, and should be kept to a minimum. Measure twice, cut once. You should consider making ample prototypes and mockups during this phase. It is your last chance to work with the design before developers methodically step through the process of implementing it. It is important to avoid situations where your developers completely implement a feature only to learn that you redesigned it later.

Issue Management

During the lifecycle of a product, thousands of issues will arise. Some will be defects, bugs, enhancement requests, customer requirements, sales dependencies, new technologies to research, and many other types. A central repository for these issues is a necessity. Some companies use text files, databases, or other more suitable tracking products like Microsoft Project Server® or Standard Issue®. Regardless of your technique, you must manage these issues. Using a product that enforces workflow will help ensure that issues are dealt with appropriately.

Quality Assurance

It is wise to have procedures for routing defects through the QA team to ensure that each issue is addressed, and nothing is swept under the rug. The QA team should triage and prioritize issues for development. When hundreds or even thousands of issues must be managed, it is a lot less painful to use a tool designed for the job. During this time, development should adopt a role of eliminating defects, customer problems, and reasons not to ship the product. QA is the driver in this effort, and development is the instrument.

Release Management

After the initial release of a product is complete, you have the burden of ensuring that each subsequent release also meets customer needs while efficiently utilizing development resources. This is a delicate balance. Development usually dictates the duration of time that each feature may take. Management can use this to strategically stack up features for each release and maintain critical release windows. Proper resource allocation to meet these windows is the trick.

Seasons of Life

For every product, there is a season for everything - a time to be born, and a time to die, a time to build, and a time to tear down. It is important to keep in mind where your product is in its own lifecycle, as well as where it is in the lifecycle of the market it serves. Even markets have lifecycle periods such as “innovators,” “early adopters,” “early majority,” “late majority,” and “laggards.” The investment you make into your product should depend upon the market cycle it lives in. For instance, you would not make large investments in architecture when your market is in decline. DOS based applications are a perfect example. You would probably never consider updating them because the markets they serve are either dead or in decline.

Developing products well requires discipline in the many areas discussed above. The application life cycle of your product must be studied and good project management principles applied to achieve success. Product development success does not come easily.

About Us

Scoutwest, Inc. develops and publishes project management and time tracking products for consulting, manufacturing, government, and general business applications.

Thousands of small to large businesses, in dozens of countries worldwide, trust their mission critical business processes to Scoutwest products. Standard Time® and Standard Issue® work together to offer well-rounded project management solutions.

We specialize in packaged software for timesheets, project management, time tracking, defect tracking, and issue tracking. Standard Time is a web-based timesheet that also runs on Windows, Palm OS, and Pocket PC. It can be used for client billing and task management. Standard Issue is used for bug tracking and general issue tracking.

Please visit these web sites for more information.

www.stdtime.com
www.sdtissue.com