SOURCEFORGE COLLABORATIVE DEVELOPMENT PROCESS

PHASE 1: PRODUCT DEFINITION

PHASE 2: DESIGN

PHASE 3: DEVELOPMENT

PHASE 4: RELEASE PREPARATION

PHASE 5: RELEASE

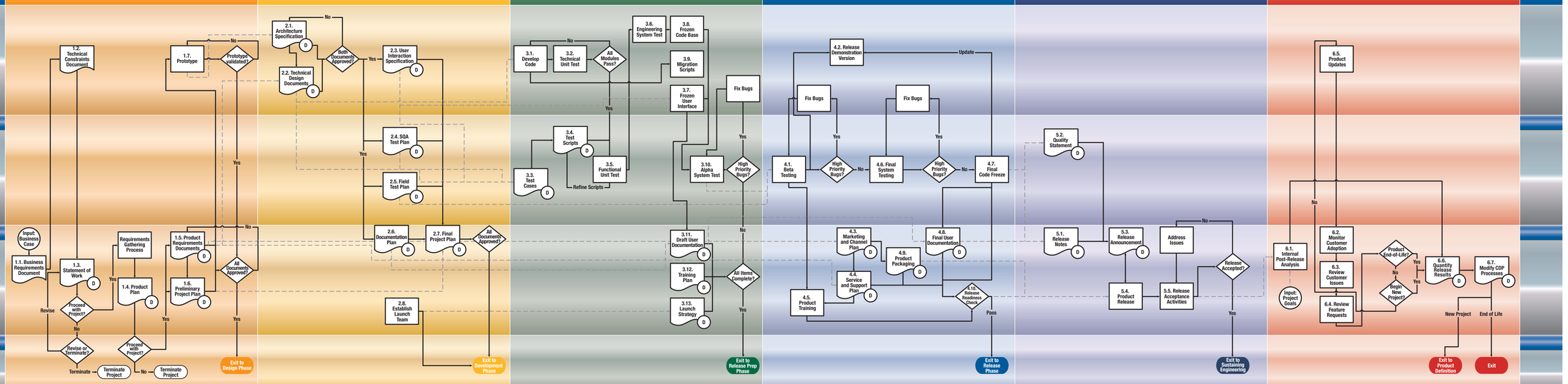
PHASE 6: SUSTAINING ENGINEERING

SOFTWARE ENGINEERING

SOFTWARE QUALITY ASSURANCE

SOFTWARE PRODUCT MANAGEMENT

OTHER



PRODUCT DEFINITION PHASE DELIVERABLES

1.1. Business Requirements Document – Describes the business objectives, market opportunity, and strategic goals of the proposed product.
1.2. Technical Constraints Document – Provides an initial technical feasibility, constraints, and effort estimate for the proposed product.
1.3. Statement of Work – Combines the Business Requirements and Technical Constraints Documents for ease of review.
1.4. Product Plan – Provides a comprehensive overview of the product and the development effort required to produce it. Includes product requirements overview, feature-level technical effort estimates, quality, cost, and delivery, risk analysis, roles and responsibilities, and key deliverables. Serves as a decision point for proceeding with the project.
1.5. Product Requirements Documents – Expand on the information provided in the Product Plan to provide detailed, feature-level requirements specifications.
1.6. Preliminary Project Plan – Provides a milestone-level project schedule. Will be expanded in later stages.
1.7. Product Prototype – Validates the product architecture and further refines the technical effort estimates.

DESIGN PHASE DELIVERABLES

2.1. Architecture Specification – Provides a detailed description of the product architecture.
2.2. Technical Design Documents – Provide detailed descriptions of the technical requirements, development methods, and implementation specifications for each product feature. Must conform to the Product Requirements Documents and Architecture Specification.
2.3. User Interaction Specification – Describes the manner in which users will interact with the product. Includes items such as task flow, navigation, coding guidelines, display of data, page layout, and textual and graphical elements.
2.4. Software Quality Assurance Test Plan – Provides a comprehensive overview of all software testing activities.
2.5. Field Test Plan – Provides an overview of all pre-release field testing activities such as Alpha and Beta testing.
2.6. Documentation Plan – Outlines all technical documentation to be produced in support of the product.
2.7. Final Project Plan – Expands on the Preliminary Project Plan to provide task-level detail.
2.8. Establish Launch Team – Establishes a cross-functional team to oversee the progress of the project.

DEVELOPMENT PHASE DELIVERABLES

3.1. Develop Code – Develop and test all product code.
3.2. Technical Unit Testing – Tests each code module individually, prior to integration or handover to Software Quality Assurance. White box testing involving code review.
3.3. Test Cases – Provide a high-level overview of the product components to be tested during software testing activities.
3.4. Test Scripts – Document the specific scenarios to be tested during software testing activities in step by step detail.
3.5. Functional Unit Testing – Tests each code module individually. Black box testing.
3.6. Engineering System Testing – Tests integrated product.
3.7. Frozen User Interface – Marks completion of user interface.
3.8. Frozen Code Base – Marks completion of all product code. Change control is initiated at this point.
3.9. Migration Scripts – Developed to enable users to upgrade from an earlier product version.
3.10. Alpha System Testing – Tests integrated product. May be opened to participants external to the organization.
3.11. Draft User Documentation – Provide drafts for review.
3.12. Training Plan – Describes all product training activities.
3.13. Launch Strategy – Describes key product launch activities.

RELEASE PREPARATION PHASE DELIVERABLES

4.1. Beta Testing – Enlists participants external to the development organization to use and test the product.
4.2. Release Demonstration Version – Enables pre-release product demonstrations and facilitates training, marketing, and documentation activities.
4.3. Marketing and Channel Plan – Documents all marketing-related activities.
4.4. Service and Support Plan – Documents the strategy for supporting the product and the activities necessary to prepare support staff.
4.5. Product Training – Trains internal staff to use, sell, and support the product.
4.6. Final System Testing – Tests the product once all bug fixes and other code changes have been completed.
4.7. Final Code Freeze – Marks the final completion of all product code. No further changes are permitted.
4.8. Final User Documentation – Complete all final user documentation.
4.9. Product Packaging – Complete packaging materials.
4.10. Release Readiness Checkpoint – Confirms organization-wide release readiness.

RELEASE PHASE DELIVERABLES

5.1. Release Notes – Provides users critical product information and any late updates to the user documentation. Includes key product enhancements, corrected defects, known issues, and references to comprehensive documentation.
5.2. Quality Statement – Summarizes the outcome of all software testing activities.
5.3. Release Announcement – Announces the product release. May be internal or external.
5.4. Product Release – Releases the product to its intended audience.
5.5. Release Acceptance Activities – Confirm acceptance of the product by a sampling of early adopters. Validate usability of product, documentation, collateral materials, and support infrastructure.

SUSTAINING ENGINEERING PHASE DELIVERABLES

6.1. Internal Post-Release Analysis – Identifies the successes and challenges of the project. Measures planned versus actual results and identifies areas for future improvement.
6.2. Monitor Customer Adoption – Monitor and facilitate customer adoption of the product.
6.3. Review Customer Issues – Review customer-reported issues and support requests to assess the quality of the product and identify possible need for remedial action.
6.4. Review Feature Requests – Review customer feature requests for potential inclusion in patches or upcoming releases.
6.5. Patch Process – Issue product updates or patches as needed to address critical customer issues.
6.6. Quantify Release Results – Prepare detailed report of all post-release analysis results. Measure progress against established goals. Use as preparation for beginning a new project and when reaching the product's end-of-life.
6.7. Modify CDP Processes – Identify areas for process improvement and implement necessary changes. Repeat this process before beginning a new project, when reaching the product's end-of-life, and as needed to reach established process improvement goals.

(D) = ITERATIVE DOCUMENT REVIEW AND APPROVAL PROCESS

The Iterative Document Review and Approval Process uses the SourceForge Document Manager to automatically route documents for review and approval. The document submitter establishes the review criteria. Reviewers then receive email notifications, review instructions, and overdue reminders if necessary. All review comments are archived in the Document Manager for later reference.

SOFTWARE ENGINEERING INSTITUTE (SEI) CAPABILITY MATURITY MODEL (CMM)

Introduction

The Software Engineering Institute (SEI) Capability Maturity Model for Software (SW-CMM) is a framework that describes the key elements of an effective software process. A key premise is that process maturity equates to the improved ability of an organization to meet cost, schedule, functionality, and product quality goals. The SW-CMM defines five process maturity levels from 1-5. Each is comprised of Key Process Areas (KPAs) defining a series of activities that, when performed consistently and effectively, help organizations meet established goals for process capability.

Using the SourceForge Collaborative Development Process (CDP)

The CDP provides a methodology that, when combined with the use of SourceForge, fully supports all of the goals associated with achieving SW-CMM Levels 2 and 3.

The following sections provide a summary of the goals associated with each KPA and the ways SourceForge and the CDP processes can be used to achieve them.

SW-CMM Level 2 KPA: Requirements Management

Product requirements are defined & controlled. A baseline is established and software plans are kept consistent with the requirements.

Use the SourceForge Tracker to manage feature requests, Tracker/SCM integration to map code to requirements, and the Document Manager to store requirements documentation.

SW-CMM Level 2 KPA: Software Project Planning

The software project plan is developed based on requirements and effort estimates. All commitments are agreed to and documented.

Use the SourceForge Project Management Console to manage your project plan and Reporting to measure the success of past estimation efforts and aid in project planning.

SW-CMM Level 2 KPA: Software Project Tracking & Oversight

Actual results are measured against the plan. Corrective actions are taken and managed to closure and the plan is adjusted if necessary.

Use the SourceForge Project Management Console to maintain visibility into and control of all deviations from your project plan, and Tracker reporting to monitor status on tracker artifacts.

SW-CMM Level 2 KPA: Software Subcontract Management

Qualified software subcontractors are selected. Commitments are mutually agreed upon, ongoing communication maintained, and progress tracked.

Use the SourceForge Project Management Console to manage actions assigned to subcontractors and Mailing Lists to maintain communication.

SW-CMM Level 2 KPA: Software Quality Assurance

QA activities are planned, performed and verified. Products are tested to established standards and noncompliance issues are addressed.

Use the SourceForge Tracker to track bug status, the Document Manager to store test plans and other documents, and Reporting to report on QA activity.

SW-CMM Level 2 KPA: Software Configuration Management

SCM activities are planned, baselined, monitored and controlled. All code changes and adherence to SCM policies are tracked.

Use the SourceForge SCM interface to browse your repository and the Tracker/SCM integration to associate code commits with bugs and requirements artifacts.

SW-CMM Level 3 KPA: Organization Process Focus

Organizational responsibility for assessing, developing, maintaining, and improving software process activities is established.

Establish a software engineering process group. Set up a SourceForge project to store all of the team's documents, files, forums, mailing lists, and other materials.

SW-CMM Level 3 KPA: Organization Process Definition

A set of standard software processes for the organization is established, maintained, and used by all software projects.

Use your SourceForge project as a central repository for all of the organization's process documents, templates, and other process library materials.

SW-CMM Level 3 KPA: Training Program

Identify and provide the training needed by the organization, projects, and individuals to perform their roles effectively.

Use the SourceForge Document Manager to store training plans and materials. Conduct product training during the Release Preparation phase of every project.

SW-CMM Level 3 KPA: Integrated Software Management

Each software project follows a set of processes tailored from the organization's standard software processes.

During each project's Product Definition and Design phases, select and tailor project-specific processes from your SourceForge process repository project.

SW-CMM Level 3 KPA: Software Product Engineering

Build and maintain the product software following the project's defined software processes.

Use the SourceForge Tracker/SCM integration with code commits, the SCM interface to browse your repository, and your project to access process documents.

SW-CMM Level 3 KPA: Intergroup Coordination

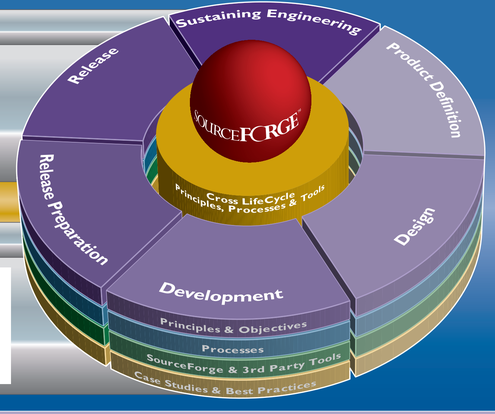
Software Engineering groups work with other engineering groups to address system-level requirements, objectives, and issues.

Use SourceForge Forums and Mailing Lists to coordinate the groups' activities and the Project Management Console to manage all actions and deliverables.

SW-CMM Level 3 KPA: Peer Reviews

Peer reviews of code are conducted to remove defects from the software early and efficiently.

Use the SourceForge SCM interface to access the code and the Document Manager to store review templates. Conduct throughout the Development phase.



VA SOFTWARE

VA Software Corporation • 47071 Bayside Parkway • Fremont, CA 94538
877.825.4689 toll free • 510.687.7000 phone • 510.683.0710 fax
http://www.vasoftware.com

SOURCEFORGE™