8. Release Management Plan

Successful software development requires central management, a centralized systematic tracking system, a centralized common code repository, regular automated builds, and a structured design, review, and testing infrastructure.

8.1. Inception and Design Assessment

In the inception phase for the development of a software component, the subproject leader first prepares a vision statement for the purpose, functionality, and acceptance criteria for the component. This document is reviewed and approved by the Project Manager, and is entered into the project tracking system as a feature request.

The subproject leader next derives functional requirements and constraints, and describes critical use cases for the task. The use cases and functional requirements are reviewed and approved by the Project Manager, then committed to a versioning repository (i.e., cvs). A class structure of the component is written without methods, but with enough completeness so component architecture descriptions can be generated by automated tools such as doxygen.

The subproject development team next produces prototypes that provide the critical functionalities. The purpose of the prototype is to clarify requirements, and
several prototypes may be built. Tests of behavior are constructed concurrently with prototyping, and are used by the Project Manager to assess how well the design can provide the required functionalities.

The Project Manager schedules a code design review with the subproject leader’s development team. A design review team includes at least the subproject leader and Project Manager, and the team may involve include developers from Central Services and other members of the DANSE community. Design patterns and UML diagrams describing component architecture are reviewed. Design approvals from the review are documented and committed to the versioning repository along with any updates to the specification and design documentation. The Project Manager edits the feature requests in the project tracking system, as appropriate.

The subproject leader and Project Manager informally reassess the task’s cost and development schedule in the WBS.

8.2. ELABORATION PHASE

In the elaboration phase, components are built from the approved design. A full product prototype with core component architecture is built to support the critical use cases. An automated certification testing suite is designed and initiated.

A critical code review is scheduled by the Project Manager to assess the restructuring of the code into more generic or reusable components, and to confirm quality standards within the code. When the code review is complete, the code is transitioned to the main release branch of the versioning repository, and the Project Manager adjusts the feature requests in the project tracking system, as appropriate. A baseline architecture has been established, and an alpha build of the component is released.

This elaboration phase may include changes to architectural design, reassessment of risk, reassessments of cost, and some change in scope. The Project Manager and subproject leader again discuss potential changes to cost and schedule, and this information is communicated to the Project Controls group as appropriate.

8.3. CONSTRUCTION PHASE

All the remaining features are designed and implemented into the component architecture. Test cases are generated simultaneously to verify all use cases, and an automated testing suite is constructed and executed for the full code. The focus of this phase is to optimize cost, effort, and product quality. A beta version of the component, now with full functionality, is released.

8.4. TRANSITIONAL PHASE

The last phase of the release cycle is entered when the component is ready to be deployed to the full target domain (i.e., platforms, operating systems, compilers). Debugging and usability improvements are performed based on acceptance testing (i.e., end-user feedback). Documentation is also expanded to the full scale, and user manuals are updated. Certification testing is completed by the Quality Assurance agency.
The component design phase is considered complete. Further development and design changes are considered as part of integrated application development.

8.5. Application Release Process

Applications are built from networks of components, and are the primary deliverables in the DANSE releases. The application release process is similar to that for components, with components treated as black boxes. Testing is different for applications, however. Two types of components exist on the pyre framework: 1) fixed scripts composed of specific components, and 2) flexible applications that utilize “facilities” selected from sets of components housed in architecture-defined locations. The process for testing fixed scripts will be similar to the testing of components. Full “facility” applications can only be tested in the most critical usage cases, however, and further confidence for certification will rely on application-level code reviews.

8.6. Release Criteria

Each DANSE release is composed of a set of individual components and integrated applications. All components and applications shall be classified by these criteria:

- *Prototype* denotes components or applications that have at least completed the inception phase.
- *Alpha* denotes components or applications that have at least completed the elaboration phase.
- *Beta* denotes components or applications that have at least completed the construction phase.
- *Numbers* such as 1.0 and 1.1 denote components or applications that have completed the transition phase.

We note that there will be a mix of components and applications in each release of the DANSE software (e.g., the beta release of the DANSE software may contain some components that have achieved 1.0 status, and some applications that have achieved alpha status). A complete plan for the components in each release, and their status of completion, is given in the DANSE Release Plan.