AUTHOR

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Department** |  | **Signature** |  | **Print Name** |  | **Date** |
| R&D |  |  |  | Ken Woodland |  |  |

SIGNATURES

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Department** |  | **Signature** |  | **Print Name** |  | **Date** |
| R&D |  |  |  |  |  |  |

Revision History

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Rev.** |  | **Date** |  | **Initials** |  | **Description** |
| 0.1 |  | 19-Oct-2018 |  | KW |  | Initial Creation |
| A |  | 26-Sep-2019 |  | DAT |  | Update to Revision A |

Glossary

ECR – Engineering Change Request

ECO – Engineering Change Order; the result of approving an ECR per the Engineering Change ECR Process.

SAP – Document Control System used by the Document Control System Procedure.

References

1. 15008077 – DYONICS POWER II Software Development Plan
2. 15008080 – DYONICS POWER II Software Description
3. 15000695 – DYONICS POWER II System Controller Software Requirements Specification
4. 15000701 – DYONICS POWER II Motor Controller Software Requirements Specification
5. 15000699 – DYONICS POWER II Software Trace Matrix
6. 15000696 – DYONICS POWER II System Controller Software Design Specification
7. 15000702 – DYONICS POWER II Motor Controller Software Design Specification
8. 15008082 – DYONICS POWER II Software Architecture Diagram
9. 16500014 – DYONICS POWER II Risk Analysis
10. 15008021 – DYONICS POWER II Software Level of Concern
11. 14000095 – Engineering Change ECR Process
12. 1400019 – Document Control System Procedure

# Software Life Cycle Development Plan

The DYONICS POWER II Software follows a traditional waterfall model of development. The development starts with requirement specifications, proceeds to design specification, then coding and testing. If during this cycle any changes to the specification occur, the process starts back at the top again. Once the code meets the specifications and passes the Verification / Validation Tests, the software is released.

Specifics of the software development flow:

1. The User Needs are agreed upon, the Software Level of Concern is reviewed and the Software Description and Software Development Plan are started.
2. Software Requirements Specification is then created from the User Needs. The top level risks are assessed and the Software Trace Matrix is started.
3. The proposed software design is documented through the creation of the Software Design Description and Software Architecture Design Chart. Also a full Device Hazard Analysis is performed.
4. Prototypes and Proof of Concepts software is evaluated to ensure a solid design.
5. The design is frozen when the requirements are complete.
6. A software verification plan is created. Verification protocols are created and all design documents are approved.
7. Formal Coding is then performed to the requirements and specifications. Code is put under source control.
8. Code Reviews are performed to ensure the code reflects the requirements, SN ASD coding standards and will work as designed.
9. Code is frozen with engineering version assigned.
10. Unit Tests are performed to verify units of the application function as designed and that risk mitigations for associated hazards are acceptable. Software Unit Test results are generated and approved. Any issues encountered during Unit Tests are recorded in a problem report.
11. Integration and System Tests are performed to verify that the software integration has been performed and that the software system functions as designed and fulfills requirements. Verification Test results are generated and approved. Issues encountered during Integration and System Tests are recorded in a problem report and reviewed for acceptability for inclusion in the final product.
12. Validation Tests are performed to determine if the software satisfies the User Needs and Intended Use. Issues encountered during Validation Tests are recorded in a problem report and reviewed for acceptability for inclusion in the final product. A Validation Test report including a final problem report is generated and approved.
13. The software and all supporting documentation are released by Engineering Control Order (ECO).

# Software Configuration Management

During the development process, software is checked in under Microsoft Visual Source Safe and assigned a unique revision. The revisions are incremented each time a new change is checked in. Software versions are labels attached to the specific revisions of files included in the product. All software and supporting documentation is checked in under the Smith & Nephew Document Control system (SAP), prior to the release and an ECO is created and electronically signed.

# Software Maintenance Plan

If a change in software is required after the release of the Software, the following steps are performed:

1. A Software Change Request is created, providing the details of the change and justification of why the change is needed.
2. The development team reviews the Software Change Request to determine the impact to software requirements, software design, risk management, and what documents to revise in order to support the change. Based upon the review the Software Change Request is approved or rejected. Approved Software Changes are then prioritized. The development team analyzes each approved change to determine the method of implementation.
3. The documents identified in Step 2 are revised to reflect the change and approved.
4. Formal coding starts when the method of implementing the change is finalized. Code written from this point forward must conform to SNE standards. Formal source control of the code starts at this step.
5. Unit, Integration and Regression Verification testing ensures that implementing the change did not affect other areas of the software and that the change works as expected.
6. Validation testing ensures that the software still satisfies the User Needs and Intended Use after the change.
7. The software and all supporting documentation are released by ECO.