# KOTDA SOFTWARE DEVELOPMENT PROCEDURE

Table of Contents

[Terminology, Definitions and Abbreviations 5](#_Toc69289592)

[Process 5](#_Toc69289593)

[General Remarks 5](#_Toc69289594)

[Process Workflow 5](#_Toc69289595)

[Process Description 8](#_Toc69289596)

[General 8](#_Toc69289597)

[Software Development Process 9](#_Toc69289598)

[Establish the Software Requirements 9](#_Toc69289599)

[Perform a Technical Review on the Software Requirements 11](#_Toc69289600)

[Initiate and Plan the Software Development 13](#_Toc69289601)

[Initiate and Plan the Software Configuration Management 16](#_Toc69289602)

[Establish the Software Test Plan 19](#_Toc69289603)

[Initiate and Plan the Software Quality Assurance 21](#_Toc69289604)

[Perform a Technical Review on the Software Planning 23](#_Toc69289605)

[Perform a S1 Management Review 25](#_Toc69289606)

[Milestone S1 26](#_Toc69289607)

[Establish the Software Design 27](#_Toc69289608)

[Update the Software Test Plan 28](#_Toc69289609)

[Perform a Technical Review on the Software Design 28](#_Toc69289610)

[Coding 30](#_Toc69289611)

[Perform a Technical Review on the Software Source Code 32](#_Toc69289612)

[Specify the Software Unit Tests 34](#_Toc69289613)

[Perform a Technical Review on the Software Unit Test Specification 37](#_Toc69289614)

[Prepare and Perform the Software Unit Tests 39](#_Toc69289615)

[Perform a S2 Management Review 40](#_Toc69289616)

[Milestone S2 42](#_Toc69289617)

[Specify the Software Integration Tests 42](#_Toc69289618)

[Perform a Technical Review on the Software Integration Test Specification. 43](#_Toc69289619)

[Prepare and Perform the Software Integration Tests 45](#_Toc69289620)

[Problem Resolution Meeting 47](#_Toc69289621)

[Specify the Software Validation Tests 48](#_Toc69289622)

[Perform a Technical Review on the Software Validation Test Specification 49](#_Toc69289623)

[Prepare and Perform the Software Validation Tests 52](#_Toc69289624)

[Problem Resolution Meeting 53](#_Toc69289625)

[Perform a S3 Management Review 55](#_Toc69289626)

[Milestone S3 56](#_Toc69289627)

[Release the Software 57](#_Toc69289628)

[Role Descriptions 58](#_Toc69289629)

[The Software Project Manager 58](#_Toc69289630)

[The Software Requirements Engineer 60](#_Toc69289631)

[The Software Developer 62](#_Toc69289632)

**Purpose: -** This document describes the overall software development process of web, desktop and mobile application software during all phases of Konza Technopolis Development Authority Software development life cycle.

**Scope: -** Valid for all web, desktop and mobile applications developed for Konza Technopolis Development Authority.

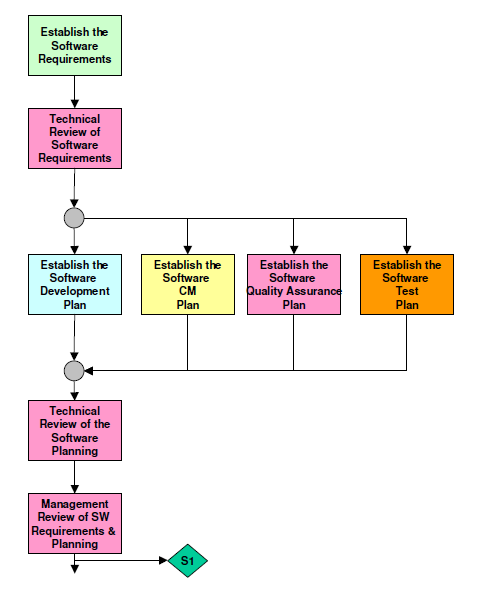
# Terminology, Definitions and Abbreviations

# Process

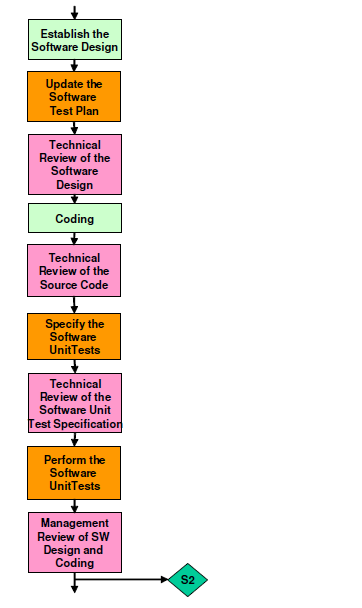
## General Remarks

*For many of the documents an object review is mentioned in this process. You must tailor this according to your needs. It is usually not possible to perform an object review on each small modification of a document. It is also not necessary to perform a code review on each software unit and on each modification of a software unit. ISO/IEC 15504 requires you to have a strategy and to document this strategy. For example, only review the initial versions of a document and only review the software units with high criticality*

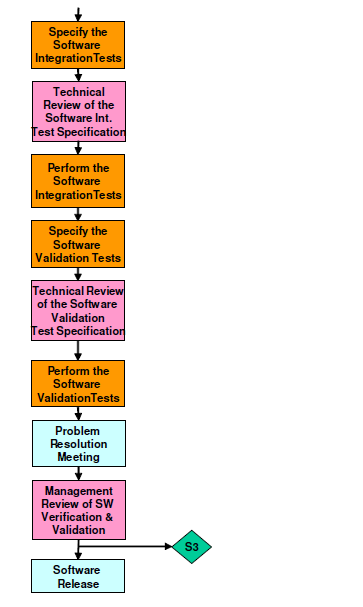
## Process Workflow

****

***Figure 1:*** *Software development until milestone S1.*

**

***Figure 2:*** *Software development until milestone S2.*

**

***Figure 3:*** *Software development until milestone S3/Software release.*

# Process Description

## General

*Describe here some general things like how the release loops are handled e.g., if it is possible to exit at any stage of the process and enter at a previous stage i.e., to fix problems prior to a moving on in the process.*

*If the software development is part of a system development process describe how the process is embedded into the overall process and how the interfaces are managed.*

# Software Development Process

## Establish the Software Requirements

|  |  |
| --- | --- |
| **Goal:** | **Establish a valid Software Requirements Specification**   * Software requirements are identified and specified * Deviations from the stakeholder’s requirements are identified and reconciled with the customer / stakeholder. |
| **Input:** | * User /Customer or other stakeholder requirements |
| **Output:** | * Valid document: **Software requirements Specification Document (.doc)** |
| **Methods and Templates:** | * Standard document: **KoTDA Software Requirements Specification Document Template (.doc) [***insert link]* * Design thinking |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Analyze stakeholder requirements documents** | Derive software requirements applying the methods as provided in the section “Methods and Templates” above. |  |  | Software Requirements Engineer |
| **Specify additional requirements** | Generally, after analyzing the stakeholder’s requirements, some requirements have not been covered or are not complete.   * Not yet covered or deviating stakeholder requirements from the previous action must be clarified with the stakeholder and included into the specifications. * Perform a proper use case analysis, considering all possible users of the system and generate scenarios for each use case to make sure that the requirements cover all needed topics. |  |  | Software Requirements Engineer |
| **Clarify open issues and assumptions** | * Clarify the open issues and assumptions from the use case analysis with the stakeholders. * Update the requirements as needed. * Make sure that the performed updates do not lead to contradictions. If this should happen, solve contradictions. |  |  | Software Requirements Engineer |
| **Collect feedback on requirements from all parties concerned** | * Contact all parties which are concerned with by the requirements (e.g., system architects, designers, representatives of the next development level) and collect their feedback regarding e.g. testability, feasibility, programmability. * Problems for individual requirements should be reconciled and fixed in the software requirements specification. |  |  | Software Requirements Engineer |

## Perform a Technical Review on the Software Requirements

|  |  |
| --- | --- |
| **Goal:** | **Establish a reviewed version of the document from section 2.3.2.1. above.**  The goal is to review the completed version of the SRSD and to evaluate its suitability for the intended use. |
| **Input:** | Completed version of SRSD from section 2.3.2.1 to be reviewed. |
| **Output:** | * Reviewed version of the work product. * Review report or comment list: * Valid document: **Software Review Checklist (.doc)** |
| **Methods and Templates:** | * **KoTDA Software Review Checklist Template(.doc)** * **[***Suggest software review method***]** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Plan and initiate the technical review** | The technical review must be performed on the document in scope. A dedicated meeting for a walkthrough is not mandatory. The reviews can be performed as peer reviews.  The owner or person responsible for the document to be reviewed must distribute the document to appropriate peer reviewers. It is mandatory to have at least two peer reviewers.  The owner of the project/object has to set the date of the review, i.e., when he expects the review results to be back. |  |  | Software Requirements Engineer |
| **Preparation by the reviewers** | The peer reviewers will carefully study the review object. The focus is to answer the related review checklist questions.  The peer reviewer must apply his experience and common sense to detect problems and errors in the reviewed SRSD which may not be covered by the review checklist questions.  The prepared checklists and comment sheets have to be ready for the review meeting or alternatively have to be handed in to the person responsible for the technical review. |  |  |  |
| **Perform the technical review meeting** | The person responsible for the technical review must call a review meeting where the peer reviewers participate to reconcile and discuss the review findings. Alternatively, he can do this without any participants, using the pre-filled comments and checklists of the peer reviewers.  The peer reviewers must agree to this report by their signature.  If the review status was “**passed**”, and the need for work was not identified, there are no further actions required. If the modifications to the SRSD are only minor, an agreement has to be reached among the reviewers and the person responsible for the SRSD, about what should be modified and the person responsible for the document without new review.  If the status of the technical review was “**failed”**, the person responsible for the technical review has to modify the SRSD according to the findings of the review. This may involve discussion and agreements with the peer reviewers until a satisfactory solution is found and implemented. The goal is that the peer reviewers can agree to the modification.  This means that the technical review must be performed again on the modified SRSD and the described procedure has to be followed until the status can be set to **“Passed”.** |  | ICT Officer (Software & Applications) |  |
| **Documentation and tracking** | The review results and related forms and documents i.e. the review checklist, comments, and status results have to be placed into the project Configuration Management archives.  The labeling of these files must be performed to relate them to the appropriate baseline.  All review findings which should lead to a modification of the review object have to be submitted to the problem solution process. |  |  | none |

## Initiate and Plan the Software Development

|  |  |
| --- | --- |
| **Goal:** | **Establish the Software Development Planning.**   * The roles and responsibilities within the project must be clearly defined at this point. The Software development plan must be established. Note that this step is in parallel to initiate and plan the configuration management and to initiate and plan the quality assurance. |
| **Input:** | SRSD in sufficient quality to allow resource estimations.  **Note:** *Requirements engineering, and planning of the software development are not sequential. They run in parallel. Some of the requirements engineering is needed to perform the planning and the bigger part of the requirements engineering needs to be planned and documented in the Software Development Plan.* |
| **Output:** | Valid document: **Software Development Plan.** |
| **Methods and Templates:** | * **[***Suggest Software Project Planning method***]** * KoTDASoftwareDevelopmentPlanTemplate. (.doc) * MeetingMinutesTemplate (.doc) * SoftwareResourceEstimationSheetTemplate (.doc) * Software Project Time Schedule (e.g. Gantt Chart) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Nominate the Software Project Manager.** | Nominate the Software Project Manager for the project. The latest version at this point in time. |  |  | ICT Officer (Software & Applications) |
| **Nominate the Software team members** | Nominate the members of the software team i.e. the developers, designers e.t.c. |  |  | ICT Officer (Software & Applications),  Software Project Manager |
| **Generate the Software Development Plan** | * Generate the Software Development Plan using the template **“KoTDASoftwareDevelopmentPlanTemplate. (.doc)**   **”.** The document should address the following items:  Project Planning: Tailoring of the procedures, risk management, development goals, deliveries and milestones, organization and responsibilities in the project. The detailed schedule should not be included here but kept in an appropriate tool. For resource estimation the template **“KoTDASoftwareResourceEstimationSheet (.doc)”** must be used. This resource estimation must be the base for the project schedule.  Project Control: Specify which measures are applied to supervise the project status and perform reporting.  Testing: Usually only a reference to an external Software Test Plan is made. Testing can only be established in its complete version after a valid Software Design Document. Therefore it is not required at this point in time. |  |  |  |
| **Perform the continuous project planning and tracking activities** | Perform the continuous project planning and tracking as described in your “Software Project Planning Method”. This should involve especially the frequent update of the time schedule.  There should be a project meeting established where the tracking of the project is done. The meeting minutes should be filled in as a documentation of the tracking. For small projects where a regular meeting seems inappropriate e.g. if only one person is assigned to the project, the action item list according to the template “**KoTDAActionItemListTemplate (.doc)**” can be used instead. |  | Software Project Manager |  |

## Initiate and Plan the Software Configuration Management

|  |  |
| --- | --- |
| **Goal:** | **Establish the Configuration Management for Software Development**   * The role of the configuration manager within the projects must be assigned at this point. * The software Configuration Management Plan must be established for the project and the infrastructure for Configuration Management must be set up, as e.g. archives and tools. |
| **Input:** | Software Development Plan (partially filled in) |
| **Output:** | * Valid document: **SoftwareConfigurationManagementPlan (.doc)** * Configuration Management Infrastructure |
| **Methods and Templates:** | * **[**Suggest Software Configuration Management Method**]** * **KoTDASoftwareConfigurationManagementPlanTemplate (.doc)** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Nominate a Software Configuration Manager** | Nominate a Configuration Manager for the project to manage and or perform all necessary Configuration Management activities.  This role may be recruited from the software developers, although it is recommended to have a dedicates Software Configuration Manager. |  | Software Project Manger |  |
| **Generate the Software Configuration Management Plan** | Change management, version control, communication, data spaces and structures, tools and their maintenance must be planned and set up.  The planning has to be documented in a separate Configuration Management plan.  **Note:***It is possible and recommendable to perform Configuration Management not on project but on organization level i.e. the archives structures, labelling and naming conventions have to be the same for all projects. In this case, it is also recommendable to have a generic Configuration Management Plan.* |  |  |  |
| **Setup the Configuration Management spaces** | Change management, version control, communication, data spaces and structures, tools and their maintenance have to be set up as described in your ‘Software Configuration Management Method’ and the Software Configuration Management Plan. |  |  |  |
| **Perform the continuous Configuration Management activities** | Perform the continuous activities of Configuration Manager as described in your ‘Software Configuration Management Method’. |  | Software Configuration Manager | None |

## Establish the Software Test Plan

|  |  |
| --- | --- |
| **Goal:** | **Establish the Software Test Plan**  The scope of the Software Test Plan is:   * Software Unit Test * Software Integration and Validation Test   The goals of the Software Unit Test:   * Approval that the software units are compliant to the Software Design Document, as well as the discipline related programming standards and rules.   The goals of the Software Integration and Validation Test:   * Approval that the different software units or components interact correctly at their interfaces as defined in the software architecture. This includes possible dynamic behavior and timing. * Approval that the software meets the requirements as defined in the Software Requirements Specification. * To achieve these goals, this task provides the basis from a test management point of view to focus on appropriate (e.g. high priority) test goals and test topics.   Realistic effort/costs estimations and scheduling for related work packages to enable their systematic performance.  Note that the test plan shall be established in parallel to the software project planning, based in the software requirements.  However, it needs to be updated after the design is established to reflect all architecture and design details. |
| **Input:** | Software Requirements Specification Document  Software Development Plan  Software Design Document |
| **Output:** | SoftwareTestPlan (.doc) |
| **Methods and Templates:** | **[**Suggest software testing method**]**  KoTDASoftwareTestPlanTemplate (.doc) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Generate the Software Test Plan** | Identify the test requirements and define the test strategy and focus for the project, include:   * Test goals * Test phases * Test objects * Test methods * Test documentation * Test end criteria   Define the integration approach and integration steps. Define the test methods, tools and environment e.g. development environment, test stubs, conditional compilation, language dependent solutions, test equipment, specific skills and trainings of the testers, e.t.c. Clarify the management of anomalies i.e. reporting and tracking.  Estimate the effort, costs and schedules of test engineering activities, including:   * Test phases and activities (resources, schedules, investments, training) * Detailed list of test objects * Development of test specifications * Provision of test environment / test data / test programs * Training * Test conduct * Documentation of tests * Review of results   Clarify the responsibilities for test engineering activities and synchronize test engineering activities with other plans e.g.:   * Software Development Plan * Overall Test Plan (Software Validation as part of Product Test)   Document the above steps in the Software Test Plan. |  | Software Project Manager | Software Tester. |

## Initiate and Plan the Software Quality Assurance

|  |  |
| --- | --- |
| **Goal:** | **Establish the Quality Assurance Plan for Software Development.**  The role of a quality planning engineer for the project must be assigned at this point. The Software Development Plan contains a chapter about quality assurance. This must be filled in. |
| **Input:** | Software Development Plan (Partially filled in) |
| **Output:** | Software Development Plan with the part of the quality assurance filled in. |
| **Methods and Templates:** | **[**Suggest your Software Project Planning Method**]** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Nominate the Software Quality Assurance Engineer** | Nominate the Software Quality Assurance Engineer for the project to manage and/or perform all quality assurance activities.  It is mandatory that the person fulfilling this role is not a member of the product development team. |  |  |  |
| **Generate the Software Quality Assurance Plan** | The software QA engineer must define the quality assurance measures applied in the project in agreement with the software Project Manager, (these are for example, the reviews, project control, status reporting and metrics). This is predefined in the section about quality assurance in the Software Development Plan and may be tailored according to the allowed tailoring measures. No separate review is defined for the quality status section. It is reviewed together with the review of the complete Software Development Plan. |  | Software QA Engineer. | Software Project Manager,  ICT Officer (Software & Applications) |
| **Perform the continuous QA activities** | Perform the continuous QQA activities. This involves a regular evaluation of the status of the project by independent quality assurance staff. The QA engineet has to document the status in the Software Quality Status Sheet and report it in a defined frequency to the Software Management team. |  | Software QA Engineer. |  |

## Perform a Technical Review on the Software Planning

|  |  |
| --- | --- |
| **Goal:** | **Establish a reviewed version of the Software Development Plan document.**  The goal is to review the completed version of the document and to evaluate its suitability for the intended use. |
| **Input:** | Complete version of the Software Development Plan to be reviewed |
| **Output:** | Reviewed version of the Software Development Plan  Review report and/or comment list: SoftwareReviewChecklist (.doc) |
| **Methods and Templates:** | * **[**Suggest Software Review Method**]** * **KoTDA Software Review Checklist Template(.doc)** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Plan and initiate the technical review** | The technical review must be performed on the document in scope. A dedicated meeting for a walkthrough is not mandatory. The reviews can be performed as peer reviews.  The owner or person responsible for the document to be reviewed must distribute the document to appropriate peer reviewers. It is mandatory to have at least two peer reviewers.  The owner of the project/object has to set the date of the review, i.e., when he expects the review results to be back. |  | Software Project Manager |  |
| **Preparation by the reviewers** | The peer reviewers will carefully study the Software Development Plan. The focus is to answer the related review checklist questions.  The peer reviewer must apply his experience and common sense to detect problems and errors in the reviewed Software Development Plan which may not be covered by the review checklist questions.  The prepared checklists and comment sheets must be ready for the review meeting or alternatively have to be handed in to the person responsible for the technical review. |  |  |  |
| **Perform the technical review meeting** | The person responsible for the technical review must call a review meeting where the peer reviewers participate to reconcile and discuss the review findings. Alternatively, he can do this without any participants, using the pre-filled comments and checklists of the peer reviewers.  The peer reviewers must agree to this report by their signature.  If the review status was “**passed**”, and the need for work was not identified, there are no further actions required. If the modifications to the Software Development Plan are only minor, an agreement must be reached among the reviewers and the person responsible for the Software Development Plan, about what should be modified and the person responsible for the document without new review.  If the status of the technical review was “**failed”**, the person responsible for the technical review must modify the Software Development Plan according to the findings of the review. This may involve discussion and agreements with the peer reviewers until a satisfactory solution is found and implemented. The goal is that the peer reviewers can agree to the modification.  This means that the technical review must be performed again on the modified SRSD and the described procedure must be followed until the status can be set to **“Passed”.** |  | ICT Officer (Software & Applications) |  |
| **Documentation and tracking** | The review results and related forms and documents i.e., the review checklist, comments, and status results must be placed into the project Configuration Management archives.  The labeling of these files must be performed to relate them to the appropriate baseline.  All review findings which should lead to a modification of the review object have to be submitted to the problem solution process. |  |  | none |

## Perform a S1 Management Review

|  |  |
| --- | --- |
| **Goal:** | **Generate an agreement to enter the next development phase**  The goal is to evaluate the completed work documents of the previous development phase and to determine their suitability for the next development steps. |
| **Input:** | Completed versions of the work documents of the previous development phase. E.g., Software Development Plan, SRSD, etc. |
| **Output:** | Review report and/or comment list: SoftwareReviewChecklist (.doc) |
| **Methods and Templates:** | * **[**Suggest Software Review Method**]** * **KoTDA Software Review Checklist Template(.doc)** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Plan and initiate the management review** | The management review must be performed on the project progress in scope.  The Software Project Manager must distribute the documents of the previous development phase to the review participants and allow enough time for preparation between the distribution and the review meeting.  The Software Project Manager must set the date of the review. |  | Software Project Manager |  |
| **Perform the management review meeting** | The Software Project Manager must call a review meeting where the project progress will be evaluated. The focus of the meeting is to confirm that the work documents of the previous development phase are present, technically reviewed and of sufficient quality to continue with the next step in the software development process. Further, the scope is to check on the project performance concerning the schedule, resources, and quality.  The reviewers must agree to this report by their signature.  If the review status was “**passed**”, and the need for work was not identified, there are no further actions required. If the modifications to the work documents are only minor, an agreement must be reached among the reviewers and the Software Project Manager, about what should be modified and the person responsible for the document without new review.  If the status of the technical review was “**failed”**, the Software Project Manager must modify the work documents according to the findings of the review. This may involve discussion and agreements with the reviewers until a satisfactory solution is found and implemented. The goal is that the reviewers can agree to the modification.  This means that the technical review must be performed again on the modified work documents and the described procedure must be followed until the status can be set to **“Passed”.** |  | ICT Officer (Software & Applications) |  |

## Milestone S1

This milestone marks the end of the requirements and planning phase for the software sub-project and the start of the detailed design phase. The successful passing of the Milestone 1 management review automatically constitutes the reaching of the S1 milestone.

## Establish the Software Design

|  |  |
| --- | --- |
| **Goal:** | **Establish the Software Design Document**  The goal is to establish the high level software architecture and design and to establish the detailed unit design for the project. |
| **Input:** | Valid Software Requirements Specification Document (SRSD) |
| **Output:** | Valid document: **“SoftwareDesignDescription (.doc/.pdf)”** |
| **Methods and Templates** | **[** suggest a software design method]  **“KoTDASoftwareDesignDescription (.doc)” [***insert link***]** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Generate the Software Design Document** | Use the valid SRSD as well as the selected design methods and generate the high-level design (architecture).  The Software Design Document must show the important interfaces and functionalities.  The modularization into functional blocks and the related control and data flow must be documented.  Also establish the detailed unit design in the Software Design Document. The software tester must be consulted to make sure that their needs are reflected in the design.  The use of design tools is permissible if the documentation (Software Design Document) is established in sufficient quality. |  | Software Developer,  ICT Officer (Software & Applications) | Software Tester |
| **Generate the Prototype Source Code** | Implement the source code of the new software complying to the high-level design and detailed unit design of the Software Design Document.  This may be the first generation of the units, to provide the empty software unit structure and development environment. It may come from a design tool as automatically generated code.  For the implementation of the source code the relevant source code templates must be used. |  | Software Developer,  ICT Officer (Software & Applications) | none |

## Update the Software Test Plan

Make sure that the Software Test Plan is updated after the design is established. At this point the module break down was performed and the detailed test planning can be performed. Make sure that the updated test plan is reviewed again.

## Perform a Technical Review on the Software Design

|  |  |
| --- | --- |
| **Goal:** | **Establish a reviewed version of the work product**  The goal is to review the completed version of the Software Design Document and to evaluate its suitability for the intended use. |
| **Input:** | Completed version of the Software Design Document |
| **Output:** | Reviewed version of the Software Design Document  Review report and comments list: SoftwareReviewChecklist (.doc) |
| **Methods and Templates:** | **[**suggest a software review method**]**  **“KoTDASoftwareReviewChecklistTemplate (.doc)”** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Plan and initiate the technical review** | The technical review must be performed on the document in scope. A dedicated meeting for a walkthrough is not mandatory. The reviews can be performed as peer reviews.  The owner or person responsible for the document to be reviewed must distribute the document to appropriate peer reviewers. It is mandatory to have at least two peer reviewers.  The owner of the project/object has to set the date of the review, i.e., when he expects the review results to be back. |  | Software Project Manager |  |
| **Preparation by the reviewers** | The peer reviewers will carefully study the Software Design Document. The focus is to answer the related review checklist questions.  The peer reviewer must apply his experience and common sense to detect problems and errors in the reviewed Software Design Document which may not be covered by the review checklist questions.  The prepared checklists and comment sheets must be ready for the review meeting or alternatively must be handed in to the person responsible for the technical review. |  | ICT Officer (Software & Applications) |  |
| **Perform the technical review meeting** | The person responsible for the technical review must call a review meeting where the peer reviewers participate to reconcile and discuss the review findings. Alternatively, he can do this without any participants, using the pre-filled comments and checklists of the peer reviewers.  The peer reviewers must agree to this report by their signature.  If the review status was “**passed**”, and the need for work was not identified, there are no further actions required. If the modifications to the Software Design Document are only minor, an agreement must be reached among the reviewers and the person responsible for the Software Design Document, about what should be modified and the person responsible for the document without new review.  If the status of the technical review was “**failed”**, the person responsible for the technical review must modify the Software Design Document according to the findings of the review. This may involve discussion and agreements with the peer reviewers until a satisfactory solution is found and implemented. The goal is that the peer reviewers can agree to the modification.  This means that the technical review must be performed again on the modified Software Design Document and the described procedure must be followed until the status can be set to **“Passed”.** |  | Software Project Manager,  Software Tester,  ICT Officer (Software & Applications) |  |
| **Documentation and tracking** | The review results and related forms and documents i.e., the review checklist, comments, and status results must be placed into the project Configuration Management archives.  The labeling of these files must be performed to relate them to the appropriate baseline.  All review findings which should lead to a modification of the review object have to be submitted to the problem solution process. |  |  | none |

## Coding

|  |  |
| --- | --- |
| **Goal:** | **Perform a source code implementation which complies to the Software source code implementation rules.**  The source code which came into existence during the early design and simulation activities most likely will not be suitable for the required platform. The available source code must be redesigned/refactored. |
| **Input:** | Valid Software Design Document  Source code of earlier design activities or prototypes |
| **Output:** | Software source code complying to the implementation rules. |
| **Methods and Templates:** | **[**Suggest software design method**]**  **[**Define programming guidelines and naming conventions**]**  Software libraries and header templates |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Perform source code implementation and refactoring** | Use the Software Design Document and implement the source code according to the specifications and descriptions in this document.  If the source code was generated in various development cycles a refactoring may be necessary to bring back to the defined standards.  For the refactoring of source code, you have to analyze the existing source code and set up test cases to verify the functionality of the software after changes in the source code.  Implement the refactoring results and check the refactored source code to make sure that the functionality is the same as in the simulation results.  Note that the coding and code refactoring is performed in various iterations during the design phases and with different scope for the various sample levels. |  | Software Project Manager,  ICT Officer (Software & Applications) | Software Developer,  Software Tester |

## Perform a Technical Review on the Software Source Code

|  |  |
| --- | --- |
| **Goal:** | **Establish a reviewed version of the source code**  The goal is to review the completed version of a work product and to evaluate its suitability for the intended use. |
| **Input:** | Completed version of the source code |
| **Output:** | Reviewed version of the source code.  Review report and/or comment list: “SoftwareReviewChecklist (.doc)” |
| **Methods and Templates:** | **[**Suggest software review method**]**  **“KoTDASoftwareReviewChecklistTemplate (.doc)”** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Plan and initiate the technical review** | The technical review must be performed on the program in scope. A dedicated meeting for a walkthrough is not mandatory. The reviews can be performed as peer reviews.  The owner or person responsible for the program to be reviewed must distribute the program to appropriate peer reviewers. It is mandatory to have at least two peer reviewers.  The owner of the project/object must set the date of the review, i.e., when he expects the review results to be back. |  | Software Project Manager |  |
| **Preparation by the reviewers** | The peer reviewers will carefully study the source code. The focus is to answer the related review checklist questions.  The peer reviewer must apply his experience and common sense to detect problems and errors in the reviewed source code which may not be covered by the review checklist questions.  The prepared checklists and comment sheets must be ready for the review meeting or alternatively must be handed in to the person responsible for the technical review. |  | ICT Officer (Software & Applications) |  |
| **Perform the technical review meeting** | The person responsible for the technical review must call a review meeting where the peer reviewers participate to reconcile and discuss the review findings. Alternatively, he can do this without any participants, using the pre-filled comments and checklists of the peer reviewers.  The peer reviewers must agree to this report by their signature.  If the review status was “**passed**”, and the need for work was not identified, there are no further actions required. If the modifications to the source code are only minor, an agreement must be reached among the reviewers and the person responsible for the source code about what should be modified and the person responsible for the document without new review.  If the status of the technical review was “**failed”**, the person responsible for the technical review must modify the source code according to the findings of the review. This may involve discussion and agreements with the peer reviewers until a satisfactory solution is found and implemented. The goal is that the peer reviewers can agree to the modification.  This means that the technical review must be performed again on the modified source code and the described procedure must be followed until the status can be set to **“Passed”.** |  | Software Project Manager,  Software Tester,  ICT Officer (Software & Applications) |  |
| **Documentation and tracking** | The review results and related forms and documents i.e., the review checklist, comments, and status results must be placed into the project Configuration Management archives.  The labeling of these files must be performed to relate them to the appropriate baseline.  All review findings which should lead to a modification of the review object have to be submitted to the problem solution process. |  |  | none |

## Specify the Software Unit Tests

|  |  |
| --- | --- |
| **Goal:** | **Specify the Software Unit Tests**  A clear specification must be made about all the tests which will be employed. The document must give detailed information to enable the tester to setup the environment for the tests and execute them test case by test case.  The expected results and the Test End Criteria must be clearly stated. |
| **Input:** | The valid Software Test Plan, the Software Design Document, containing the detailed design of the units to be tested. |
| **Output:** | SoftwareUnitTestSpecification (.doc) |
| **Methods and Templates:** | **[**Suggest a Software Testing Method**]**  **KoTDASoftwareUnitTestSpecificationTemplate (.doc)** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Define the Static Tests and document them in the Software Unit Test Specification** | Define the relevant test cases to achieve the required static test coverage for:   * The standard automatic code checks. These may be adapted to specific items as e.g., the virtual machine used in the project to be tested. * The code inspections.   Document the test cases in the Software Unit Test Specification.  Define the End of Test Criteria for the test cases and document them in the Software Unit Test Specification document.  Make sure that the related test objects are clearly identified. |  |  |  |
| **Define the Dynamic Tests and document them in the Software Unit Test Specification** | Specify test cases for the dynamic unit test focusing on correctness of functions and groups of functions or components, as they are defined in the Software Design Document.  The tests shall include a check for robustness against unexpected against unexpected input values, memory access errors, or wrong input ranges, etc.  Define the End of Test Criteria for the test cases and document them in the Software Unit Test Specification document.  Make sure that the related test objects are clearly identified. |  | Software Project Manager | Software Tester |

## Perform a Technical Review on the Software Unit Test Specification

|  |  |
| --- | --- |
| **Goal:** | **Establish a reviewed version of the work product(software)**  The goal is to review the completed version of a work product an to evaluate its suitability for the intended use. |
| **Input:** | Complete version of the work product to be reviewed. |
| **Output:** | Reviewed version of the work product.  Review report and/or comment list: SoftwareReviewChecklist (.doc) |
| **Methods and Templates:** | **[**Suggest a software review method**]**  **“KoTDASoftwareReviewChecklistTemplate(.doc)”** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Plan and initiate the technical review** | The technical review must be performed on the program in scope. A dedicated meeting for a walkthrough is not mandatory. The reviews can be performed as peer reviews.  The owner or person responsible for the program to be reviewed must distribute the program to appropriate peer reviewers. It is mandatory to have at least two peer reviewers.  The owner of the project/object must set the date of the review, i.e., when he expects the review results to be back. |  | Software Project Manager |  |
| **Preparation by the reviewers** | The peer reviewers will carefully study the source code. The focus is to answer the related review checklist questions.  The peer reviewer must apply his experience and common sense to detect problems and errors in the reviewed source code which may not be covered by the review checklist questions.  The prepared checklists and comment sheets must be ready for the review meeting or alternatively must be handed in to the person responsible for the technical review. |  | ICT Officer (Software & Applications) |  |
| **Perform the technical review meeting** | The person responsible for the technical review must call a review meeting where the peer reviewers participate to reconcile and discuss the review findings. Alternatively, he can do this without any participants, using the pre-filled comments and checklists of the peer reviewers.  The peer reviewers must agree to this report by their signature.  If the review status was “**passed**”, and the need for work was not identified, there are no further actions required. If the modifications to the source code are only minor, an agreement must be reached among the reviewers and the person responsible for the source code about what should be modified and the person responsible for the document without new review.  If the status of the technical review was “**failed”**, the person responsible for the technical review must modify the source code according to the findings of the review. This may involve discussion and agreements with the peer reviewers until a satisfactory solution is found and implemented. The goal is that the peer reviewers can agree to the modification.  This means that the technical review must be performed again on the modified source code and the described procedure must be followed until the status can be set to **“Passed”.** |  | Software Project Manager,  Software Tester,  ICT Officer (Software & Applications) |  |
| **Documentation and tracking** | The review results and related forms and documents i.e., the review checklist, comments, and status results must be placed into the project Configuration Management archives.  The labeling of these files must be performed to relate them to the appropriate baseline.  All review findings which should lead to a modification of the reviewed program have to be submitted to the problem solution process. |  |  | none |

## Prepare and Perform the Software Unit Tests

|  |  |
| --- | --- |
| **Goal:** | **Prepare and perform the Software Unit Tests**  The specific goals of this tasks include:   * Approval that each unit is compliant to its underlying documentation. * Appropriate coverage of detailed design by test cases. * Check of the robustness of the code e.g., against boundary and overflow/underflow problems. * Check of the interfaces of functions and components. * Generation of a set of regression tests for the functions and components of the software. * Confidence that each unit is of ‘good’ quality.   Scope is the current Software version under development. |
| **Input:** | * The valid Software Test Plan * Valid Software Unit Test Specification * Test objects (units) |
| **Output:** | * SoftwareUnitTestReport (.doc) * Test objects (units) |
| **Methods and Templates:** | **[**suggest a Software Testing Method**]**  **“KoTDASoftwareUnitTestReportTemplate (.doc)”** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Perform the Static Unit Tests** | Install and configure the test and measurement tools which are used to support the static unit tests.  Perform the static unit tests and code inspection until the End of Test Criteria are reached as defined in the Software Unit Test Specification.  Make sure that the related test objects are clearly identified, and the task is properly commissioned to the test group by a filled in commissioning document. |  | Software Tester | none |
| **Perform the Dynamic Unit Tests** | Buildup / install the test environment for dynamic tests, e.g. the PHP script environment or tools to measure the code coverage. Run the tests using the methods and input data until the defined End of Test Criteria are reached as defined in the Software Unit Test Specification document.  Perform the code coverage tests as specified. Record the output data for each test case (test log).  Compare the output data with the expected output data. Make sure that the related test objects are clearly identified, and the task is appropriately commissioned to the test group by a filled commissioning document. |  | Software Tester |  |
| **Generate a test report** | Write the Software Unit Test Report and fill in the details about the found anomalies. In case of detected errors, enter the relevant data into the defect database. |  | Software Tester |  |

## Perform a S2 Management Review

|  |  |
| --- | --- |
| **Goal:** | **Generate an agreement to enter the next development phase**  The goal is to evaluate the completed work products of the previous development phase and to determine their suitability for the next development steps. |
| **Input:** | Completed versions of the work products of the previous development phase. |
| **Output:** | Review report and/or comment list: SoftwareReviewChecklist (.doc) |
| **Methods and Templates:** | **[**Suggest a Software Review Method**]**  **“KoTDASoftwareReviewChecklistTemplate (.doc)”** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Plan and initiate the management review** | The management review must be performed on the project progress of the project in scope.  The Software Project Manager must distribute the documents of the previous development phase to the review participants and allow enough time for preparation between the distribution and the review meeting. The Software Project Manager must set the date of the review. |  |  |  |
| **Perform the management review meeting** | The Software Project manger must call a review meeting where the project progress will be evaluated. The focus in the meeting is to confirm that the work products of the previous development phase are present, technically reviewed and of sufficient quality to continue with the next step in the software development process.  Further, the scope is to check on the project performance concerning the schedule, resources, and quality.  The participants of the review must set a status of the review i.e., “**passed**” or “**failed**”. The participants must agree to this report by their signature.  If the status of the management review was “**failed**”, the Software Project Manager must solve the found problems and call for a repeated review. |  |  |  |

## Milestone S2

This milestone marks the end of the design and implementation phase for the software project and start of the verification and validation phase. The Successful passing of the S2 management review automatically constitutes the reaching of the S2 milestone.

## Specify the Software Integration Tests

|  |  |
| --- | --- |
| **Goal:** | **Specify the Software Integration Tests**  A clear specification must be made about all tests which will be employed. The document must give detailed information to enable the tester to setup the environment for the tests and execute them test case by test case. The expected results and the Test End Criteria must be clearly stated. |
| **Input:** | * The valid Software Test Plan * The valid Software Requirements Specification Document |
| **Output:** | Valid document: **SoftwareIntegrationTestSpecification (.doc)** |
| **Methods and Templates:** | **[**Specify a software testing method**]**  **“KoTDA SoftwareIntegrationTestSpecificationTemplate (.doc)”** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Generate the Software Integration and Validation Test Specification** | Define the relevant test cases to achieve the required test coverage to ensure that the software meets the requirements, i.e., regarding functionality, performance, internal and external software interfaces, non-functional requirements, i.e., regarding functionality, performance, internal and external software interfaces, non-functional requirements, stress/load issues.  Define the End of Test Criteria for the test cases and document them in the Software Integration Test Specification Document. Make sure that the related test objects are clearly identified. |  | Software Project Manager | Software Tester |

## Perform a Technical Review on the Software Integration Test Specification.

|  |  |
| --- | --- |
| **Goal:** | **Establish a reviewed version of the work product**  The goal is to review the completed version of a work product and to evaluate its suitability for the intended use. |
| **Input:** | Completed version of the work product to be reviewed. |
| **Output:** | Reviewed version of the work product  Review report and/or comment list: SoftwareReviewChecklist (.doc). |
| **Methods and Templates:** | **[**Suggest a Software Review Method**]**  **“KoTDASoftwareReviewChecklistTemplate (.doc)”** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Plan and initiate the technical review** | The technical review (document review) must be performed on the program in scope. A dedicated meeting for a walkthrough is not mandatory. The reviews can be performed as peer reviews.  The owner or person responsible for the program to be reviewed must distribute the program to appropriate peer reviewers. It is mandatory to have at least two peer reviewers.  The owner of the project/object must set the date of the review, i.e., when he expects the review results to be back. |  | Software Project Manager |  |
| **Preparation by the reviewers** | The peer reviewers will carefully study the source code. The focus is to answer the related review checklist questions.  The peer reviewer must apply his experience and common sense to detect problems and errors in the reviewed source code which may not be covered by the review checklist questions.  The prepared checklists and comment sheets must be ready for the review meeting or alternatively must be handed in to the person responsible for the technical review. |  | ICT Officer (Software & Applications) |  |
| **Perform the technical review meeting** | The person responsible for the technical review must call a review meeting where the peer reviewers participate to reconcile and discuss the review findings. Alternatively, he can do this without any participants, using the pre-filled comments and checklists of the peer reviewers.  The peer reviewers must agree to this report by their signature.  If the review status was “**passed**”, and the need for work was not identified, there are no further actions required. If the modifications to the source code are only minor, an agreement must be reached among the reviewers and the person responsible for the source code about what should be modified and the person responsible for the document without new review.  If the status of the technical review was “**failed”**, the person responsible for the technical review must modify the source code according to the findings of the review. This may involve discussion and agreements with the peer reviewers until a satisfactory solution is found and implemented. The goal is that the peer reviewers can agree to the modification.  This means that the technical review must be performed again on the modified source code and the described procedure must be followed until the status can be set to **“Passed”.** |  | Software Project Manager,  Software Tester,  ICT Officer (Software & Applications) |  |
| **Documentation and tracking** | The review results and related forms and documents i.e., the review checklist, comments, and status results must be placed into the project Configuration Management archives.  The labeling of these files must be performed to relate them to the appropriate baseline.  All review findings which should lead to a modification of the reviewed program have to be submitted to the problem solution process. |  |  | none |

## Prepare and Perform the Software Integration Tests

|  |  |
| --- | --- |
| **Goal:** | **Prepare and Perform the Software Integration Tests**  The specific goals of this task include:   * Appropriate coverage of the relevant requirements and architecture elements by test cases * Generation of a set of regression tests for the functionality of the Software   Scope is the current software version under development |
| **Input:** | * The valid Software Test Plan * The valid Software Integration Test Specification * Test objects (program units) |
| **Output:** | * Valid document: SoftwareIntegrationTestReport (.doc) * Verified test objects * Test logs |
| **Methods and Templates:** | **[**Suggest a software testing method**]**  **KoTDASoftwareIntegrationTestReportTemplate (.doc)** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Perform the Software Integration and Validation Tests** | Install and configure the test environment and tools which are used to support the integration and functional tests.  Perform the tests and code inspection until the End of Test Criteria are reached as defined in the Software Integration Test Specification.  Make sure that the related test objects are clearly identified, and the task is properly commissioned to the test group by a filled in commissioning document. |  |  |  |
| **Generate a test report** | Document the test results in the Software Integration Test Report and in case of detected errors; enter the relevant data in the defect database. |  | Software Tester |  |

## Problem Resolution Meeting

|  |  |
| --- | --- |
| **Goal:** | **Plan the implementation of the bug fixes found in testing**  The goal is to discuss the test findings, evaluate their criticality and decide on the time frame for the bug fixes. This is especially necessary if the tests are performed by a separate test group. |
| **Input:** | Any of the test reports with current issues |
| **Output:** | Entries in the problem resolution database or Action Item List of the project. |
| **Methods and Templates:** | **“KoTDAActionItemListTemplate (.doc)”** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Discuss and plan the bug fixes** | The meeting has the aim to review the test report(s) and derive actions from it. The tester must explain why he considers portions in the code as defects, and he shall explain their criticality.  The project members must decide which bugs will be fixed in which release. The findings from the test shall be carried over into the problem resolution database or the project’s Action Item List.  Bugs may be grouped together when they are entered in the database.  It is recommended to make the grouping according to the possible and planned implementation of the bug fixes and the planned releases.  In case of no findings in the test reports the problem, resolution meeting can be omitted. |  | Software Project Manager | Software Developer,  Software Tester |

## Specify the Software Validation Tests

|  |  |
| --- | --- |
| **Goal:** | **Specify the Functional Software Tests**  A clear specification must be made about all tests which will be employed. The document must give detailed information to enable the tester to setup the environment for the tests and execute them test case by test case.  The expected results and the Test End Criteria must be clearly stated. |
| **Input:** | * Valid Software Test Plan, * Valid Software Requirements Specification document |
| **Output:** | Valid document: SoftwareValidationTestSpecification (.doc) |
| **Methods and Templates:** | **[**Specify Software Testing Method**]**  **“KoTDASoftwareValidationTestSpecificationTemplate (.doc)”** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Generate the Software Integration and Validation Test Specification** | Define the relevant test cases to achieve the required test coverage to ensure that the software meets the requirements, i.e., regarding functionality, performance, external software interfaces, non-functional requirements, and stress/load issues.  Define the End of Test Criteria for the test cases and document them in the Software Integration and Validation Test Specification. Make sure that the related test objects are clearly identified. |  |  |  |

## Perform a Technical Review on the Software Validation Test Specification

|  |  |
| --- | --- |
| **Goal:** | **Establish a reviewed version of the work product**  The goal is to review the completed version of a work product and to evaluate its suitability for the intended use. |
| **Input:** | Completed version of the work product to be reviewed |
| **Output:** | Reviewed version of the work product  Reviewed report and/or comment list: SoftwareReviewChecklist (.doc) |
| **Methods and Templates:** | **[**Suggest software review method**]**  **“KoTDASoftwareReviewChecklistTemplate (.doc)”** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Plan and initiate the technical review** | The technical review (document review) must be performed on the program in scope. A dedicated meeting for a walkthrough is not mandatory. The reviews can be performed as peer reviews.  The owner or person responsible for the program to be reviewed must distribute the program to appropriate peer reviewers. It is mandatory to have at least two peer reviewers.  The owner of the project/object must set the date of the review, i.e., when he expects the review results to be back. |  | Software Project Manager |  |
| **Preparation by the reviewers** | The peer reviewers will carefully study the source code. The focus is to answer the related review checklist questions.  The peer reviewer must apply his experience and common sense to detect problems and errors in the reviewed source code which may not be covered by the review checklist questions.  The prepared checklists and comment sheets must be ready for the review meeting or alternatively must be handed in to the person responsible for the technical review. |  | ICT Officer (Software & Applications) |  |
| **Perform the technical review meeting** | The person responsible for the technical review must call a review meeting where the peer reviewers participate to reconcile and discuss the review findings. Alternatively, he can do this without any participants, using the pre-filled comments and checklists of the peer reviewers.  The peer reviewers must agree to this report by their signature.  If the review status was “**passed**”, and the need for work was not identified, there are no further actions required. If the modifications to the source code are only minor, an agreement must be reached among the reviewers and the person responsible for the source code about what should be modified and the person responsible for the document without new review.  If the status of the technical review was “**failed”**, the person responsible for the technical review must modify the source code according to the findings of the review. This may involve discussion and agreements with the peer reviewers until a satisfactory solution is found and implemented. The goal is that the peer reviewers can agree to the modification.  This means that the technical review must be performed again on the modified source code and the described procedure must be followed until the status can be set to **“Passed”.** |  | Software Project Manager,  Software Tester,  ICT Officer (Software & Applications) |  |
| **Documentation and tracking** | The review results and related forms and documents i.e., the review checklist, comments, and status results must be placed into the project Configuration Management archives.  The labeling of these files must be performed to relate them to the appropriate baseline.  All review findings which should lead to a modification of the reviewed program have to be submitted to the problem solution process. |  |  | none |

## Prepare and Perform the Software Validation Tests

|  |  |
| --- | --- |
| **Goal:** | **Prepare and perform the Functional Software Tests**  The specific goals of this tasks include:   * Appropriate coverage of the requirements (especially user requirements) by test cases. * Approval that the software is compliant to the requirements. * Generation of a set of regression tests for the functionality of the software   Scope is the current software version under development. |
| **Input:** | * Valid Software Test Plan * Valid Software Validation Test Specification * Test objects (software units) |
| **Output:** | Valid document: SoftwareValidationTestReport (.doc), verified test objects, test logs |
| **Methods and Templates:** | **[**suggest software testing method**]2**  **“KoTDASoftwareValidationTestReportTemplate (.doc)”** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Perform the Software Integration and Validation Tests** | Install and configure the test environment and tools which are used to support the integration and functional tests.  Perform the tests until the End of Test Criteria are reached as defined in the Software Validation Test Specification.  Record the output data with the expected output data for all test cases (test log). Compare the output data with the expected output data.  Make sure that the related test objects are clearly identified, and the task is properly commissioned to the test group by a filled in commissioning document. |  |  |  |
| **Generate a test report** | Document the test results in the Software Validation Test Report and in case of detected errors; enter the relevant data in the defect database. |  | Software Tester |  |

## Problem Resolution Meeting

|  |  |
| --- | --- |
| **Goal:** | **Plan the implementation of bug fixes found in testing**  The goal is to discuss the test findings, evaluate their criticality and decide on a time frame for the bug fixes. This is especially necessary if the tests are performed by a separate test group. |
| **Input:** | Any of the current test reports. |
| **Output:** | Entries in the problem resolution database or Action Item List of the project. |
| **Methods and Templates:** | **“KoTDAActionItemListTemplate (.doc)”** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Discuss and plan the bug fixes** | The meeting has the aim to review the test report(s) and derive actions from it. The tester must explain why he considers portions in the code as defects, and he shall explain their criticality.  The project members must decide which bugs will be fixed in which release. The findings from the test shall be carried over into the problem resolution database or the project’s Action Item List.  Bugs may be grouped together when they are entered in the database.  It is recommended to make the grouping according to the possible and planned implementation of the bug fixes and the planned releases.  In case of no findings in the test reports the problem, resolution meeting can be omitted. |  | Software Project Manager | Software Developer,  Software Tester |

## Perform a S3 Management Review

|  |  |
| --- | --- |
| **Goal:** | **Generate an agreement to proceed with the software release**  The goal is to evaluate the completed work products of the previous development phase and to determine their suitability for the software release. |
| **Input:** | Completed versions of the work products of the previous development phase |
| **Output:** | Review report and comment list: SoftwarReviewChecklist (.doc) |
| **Methods and Templates:** | **[**Suggest software review method**]**  **“KoTDASoftwareReviewChecklistTemplate (.doc)”** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Plan and initiate the management review** | The management review must be performed on the project progress of the project in scope.  The Software Project Manager must distribute the documents of the previous development phase to the review participants and allow enough time for preparation between the distribution and the review meeting. The Software Project Manager must set the date of the review. |  |  |  |
| **Perform the management review meeting** | The Software Project manager must call a review meeting where the project progress will be evaluated. The focus in the meeting is to confirm that the work products of the previous development phase are present, technically reviewed and of sufficient quality to continue with the next step in the software development process.  Further, the scope is to check on the project performance concerning the schedule, resources, and quality.  The participants of the review must set a status of the review i.e., “**passed**” or “**failed**”. The participants must agree to this report by their signature.  If the status of the management review was “**failed**”, the Software Project Manager must solve the found problems and call for a repeated review. |  |  |  |

## Milestone S3

This milestone marks the end of the verification and validation phase for the software project. The successful passing of the S3 management review automatically constitutes the reaching of the S3 milestone. The milestone also marks the end of the software development.

## Release the Software

|  |  |
| --- | --- |
| **Goal:** | **Officially release the software**  By this step the software is officially released for use by the user. |
| **Input:** | Milestone review report for the S3 review, with the status **“Passed”** |
| **Output:** | Filled in **SoftwareReleaseForm (.doc)** |
| **Methods and Templates:** | **“KoTDASoftwareReleaseFormTemplate (.doc)”** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | **Action Description** | **Possible Tailoring**  **(***Describe here in which cases and how the process can be tailored according to project needs.***)** | **Responsible** | **Participants** |
| **Release the Software** | Check the management review report for the S3 review if it has the status “**passed**”. Fill in the Software Release Form and distribute it according to the defined distribution list. |  | Software Project Manager,  ICT Officer (Software & Applications) |  |

# Role Descriptions

## The Software Project Manager

|  |  |
| --- | --- |
| **Role** | **Software Project Manager** |
| **Reports to** | The Software Project Manager reports to the Project Manager of a customer/user project for all technical issues related to the specific customer/user project. The Software Project Manager reports to the Software management for technical issues related to a generic project. |
| **Responsibilities and purpose** | The Software Project Manager shall plan, conduct, track and control the generic Software project or the software subproject within a development project.  In case of a generic Software project the Software Project Manager is responsible for the successful conduct of the software project in terms of time, efforts, and quality.  In case of a development project for a customer/user the Software Project Manager is the main interface to the overall Project Manager and is responsible for the successful conduct of the software subproject in terms of time, efforts, and quality. |
| **Preconditions (Inputs)** | The Software Project Manager shall base the schedule of his activities on the Master Project Plan of the customer project which shows clearly defined milestones and customer/user deliveries. In case of a generic software project there is no Master Project Plan, and the schedule shall be based on the defined milestones and deliveries as agreed with the management at project set up.  The Software Project Manager shall base his planning on a resource assignment which was clearly agreed with the SW management.  The Software Project Manager shall be clearly informed about the software development process which shall be applied for the project.  The Software Project Manager shall be clearly informed about the software reuse and reusability goals which shall be applied for the project.  The Software Project Manager shall be clearly informed about the software maintainability goals which shall be applied for the project.  The Software Project Manager shall be clearly informed about other goals (as e.g., support of discipline related activities) which shall be accomplished by him. |
| **Deliverables (Outputs)** | The Software Project Manager shall generate and maintain the project specific Software Development Plan.  The Software Project Manager shall coordinate and be responsible for all software related activities in a project.  The Software Project Manager shall be responsible to deliver the required software source codes and intermediate work products (e.g. documents) in the required quality as defined in the software development process. |
| **Quality and Process tasks** | The Software Project Manager shall support and implement software development process improvement measures as defined by the software management.  The Software Project Manager shall follow the instructions defined in the company wide as well as software related guidelines and procedures.  The Software Project Manager shall follow the instructions defined in the Software Procedures, Software Quality Assurance Plan, Software Configuration Management Plan and all other binding guidelines as defined for the project.  The Software Project Manager shall closely cooperate with the Software Quality Engineer to define and apply the project quality assurance measures in line with the applicable guidelines. |
| **Standard tasks** | The Software Project Manager shall perform the team building and staffing for the software project.  The Software Project Manager shall perform the effort estimation and time scheduling for the software project. In case the SW project is a sub- project of a customer/user project this must be coordinated with the Project Manager.  The Software Project Manager shall perform the tracking of the software project's progress and perform the project controlling and reporting.  In case of a customer project the Software Project Manager shall contribute to the customer project's risk management.  The Software Project Manager has the rights to access sufficient and adequate resources to achieve his defined goals, the possibility to escalate risks where necessary and to perform an independent and realistic reporting. |
| **Skills and knowledge required** | The Software Project Manager must have detailed project management skills for software development projects.  The Software Project Manager must have leadership and communication skills to be able to coordinate activities and focus on the customer's satisfaction.  The Software Project Manager must have deep understanding and knowledge of the software development process.  The Software Project Manager must have knowledge about the technical details of the product for which the software is generated.  The Software Project Manager must have detailed software discipline understanding. |
| **Remarks** |  |

## The Software Requirements Engineer

|  |  |
| --- | --- |
| **Role** | **Software Requirements Engineer** |
| **Reports to** | The Software Requirements Engineer reports to the Software Project Manager for all technical issues. |
| **Responsibilities and purpose** | The Software Requirements Engineer shall perform the requirements engineering for the software project. |
| **Preconditions (Inputs)** | The Software Requirements Engineer shall base the schedule of his activities on the project plan of the software project and perform the related requirements engineering activities.  The Software Requirements Engineer shall be clearly informed about the software development process which shall be applied for the project.  The Software Requirements Engineer shall be clearly informed about the software project and software discipline goals. |
| **Deliverables (Outputs)** | The Software Requirements Engineer shall generate and maintain the project specific Software Requirements Specification.  The Software Requirements Engineer shall participate in the review meetings which are related to the requirements engineering phase of the project. |
| **Quality and Process tasks** | The Software Requirements Engineer shall follow the instructions defined in the company wide guidelines and procedures.  The Software Requirements Engineer shall follow the instructions defined in the Software Procedures, Software Requirements Engineering methods and guidelines and all other binding guidelines as defined for the project. |
| **Standard tasks** | The Software Requirements Engineer shall collect all available requirements for the project as e.g. customer requirements, quality and legal requirements etc.  The Software Requirements Engineer shall collect requirements and analyze them. The goal is to improve their overall quality and make them complete and unambiguous.  The Software Requirements Engineer shall generate and maintain the software specific requirements specification for the project.  The Software Requirements Engineer shall participate in review meetings related to the requirements engineering phase of the project as defined in the SW procedures. |
| **Skills and knowledge required** | The Software Requirements Engineer must have detailed understanding and knowledge of the SW development process.  The Software Requirements Engineer must have deep understanding and knowledge about requirements engineering methods and tools.  The Software Requirements Engineer must have detailed software discipline understanding. |
| **Remarks** |  |

## The Software Developer

|  |  |
| --- | --- |
| **Role** | **Software Developer** |
| **Reports to** | The Software Developer reports to the Software Project Manager for all technical issues. |
| **Responsibilities and purpose** | The Software Developer shall be responsible to develop new software or perform the necessary changes to existing software to make it suitable for use in a customer project. |
| **Preconditions (Inputs)** | The Software Developer shall base the schedule of his activities on the project specific plan.  The Software Developer shall be clearly informed about the software development process which shall be applied for the project(s).  The Software Developer shall be clearly informed about the technical goals which shall be applied for the project(s).  The Software Developer shall be clearly informed about other goals (as e.g.  support of discipline related activities) which shall be accomplished by him. |
| **Deliverables (Outputs)** | The Software Developer shall generate the source code for a software according to the technical needs or customer requirements.  The Software Developer shall generate the documentation related to the software as defined in the development process. This is especially the design document. |
| **Quality and Process tasks** | The Software Developer shall support and implement software process improvement measures as defined by the software discipline management.  The Software Developer shall follow the instructions defined in the company wide as well as software related guidelines and procedures.  The Software Developer shall follow the instructions defined in the Software Development Plan, Software Quality Assurance Plan, Software Configuration Management Plan, and all other binding guidelines as defined for the project(s). |
| **Standard tasks** | The Software Developer shall participate in the necessary requirements engineering for software.  The Software Developer shall perform the software design according to the related design standards to satisfy the requirement for the software in the best possible way.  The Software Developer shall establish the software design document to document this software design.  The Software Developer shall establish the source code according to the related design and coding guidelines.  The Software Developer is responsible to support or perform the necessary tests for a software prior to its use in a sample or production delivery. |
| **Skills and knowledge required** | The Software Developer must have deep understanding and knowledge of signal processing.  The Software Developer must have detailed knowledge of vehicle restraint systems and the involved physical principles.  The Software Developer must have detailed software design and programming skills.  The Software Developer must have detailed understanding and skills to use the tools necessary to perform software development.  The Software Developer must have knowledge about the technical details of the product for which the software is established.  The Software Developer must have detailed software discipline understanding. |
| **Remarks** |  |

* + - 1. **Software Tester**

|  |  |
| --- | --- |
| **Role** | **Software Tester** |
| **Reports to** | The Software Tester reports to the Software Project Manager for all technical issues. |
| **Responsibilities and purpose** | The Software Tester shall generate or participate in the generation of project specific test specifications and other test related documents.  The Software Tester shall execute the defined tests. |
| **Preconditions (Inputs)** | The Software Tester shall base the schedule of his activities on the project specific test plans which are part of the overall planning for the test team.  The Software Tester shall be clearly informed about the software development process which shall be applied for the project(s).  The Software Tester shall be clearly informed about the software reuse and reusability goals which shall be applied for the project(s).  The Software Tester shall be clearly informed about the software maintainability goals which shall be applied for the project(s).  The Software Tester shall be clearly informed about other goals (as e.g. support of discipline related activities) which shall be accomplished by him. |
| **Deliverables (Outputs)** | The Software Tester shall generate and maintain project specific test specifications and related test software/scripts for regression tests.  The Software Tester shall record the test results and support in the generation of test reports. |
| **Quality and Process tasks** | The Software Tester shall support and implement software process improvement measures as defined by the software discipline management.  The Software Tester shall follow the instructions defined in the company wide as well as software related guidelines and procedures.  The Software Tester shall follow the instructions defined in the Software Development Plan, Software Quality Assurance Plan, Software Configuration Management Plan, and all other binding guidelines as defined for the project(s). |
| **Standard tasks** | The Software Tester shall support the definition of a test concept for the project which includes static tests (inspections and automatic code checker), and dynamic tests.  The Software Tester shall support the Software Project Manager in the generation of plans and schedules.  Software Tester shall generate test specifications and tests (regression tests) which are in line with the product and process requirements. Especially these are unit tests, software integration tests and software validation tests.  The Software Tester shall set up the test environment in which the various tests are performed.  The Software Tester shall execute the defined tests which follow strictly established test criteria and clearly defined test cases.  The Software Tester shall record the project related test results in the defined manner.  The Software Tester shall support the evaluation and selection of test tools.  The Software Tester shall perform the configuration and application (e.g., code checker settings) of test tools.  The Software Tester shall support the project team in analyzing the errors found/spotted by the customer/user. |
| **Skills and knowledge required** | The Software Tester must have deep understanding and knowledge of test methods.  The Software Tester must have detailed programming skills for the programming language which is applied in the project.  The Software Tester must have knowledge about the technical details of the product for which the software is generated.  The Software Tester must have detailed software discipline understanding. |
| **Remarks** | In the workflow for the software tests the software tester takes a considerable part. This means that must define and implement test cases, configure test tools, and program test stubs. This requires a solid training about test methods and tools.  Preferably the tester must be dedicated to these test activities. It is not recommendable to scatter the test activities to normal software developers because the effectiveness of the tests is strongly related to the experience and dedication of the testers. |