

**<Project Short Name>**

**<Project Full Name>  
<Customer>**

# Software Development Plan

# Software Development Plan

Company Logo

Department

---

Version	Document maturity (draft / valid)	Date of Issue (200x-MM-DD)	Author/Owner	Check/Release	Description
0.1	draft				Initial version

**TABLE OF CONTENTS**

**1 INTRODUCTION .....5**

1.1 PURPOSE ..... **FEHLER! TEXTMARKE NICHT DEFINIERT.**

1.2 SCOPE ..... **FEHLER! TEXTMARKE NICHT DEFINIERT.**

1.3 DOCUMENT RELATED DEFINITIONS ..... **FEHLER! TEXTMARKE NICHT DEFINIERT.**

    1.3.1 *Definitions and Terminology* ..... *Fehler! Textmarke nicht definiert.*

    1.3.2 *Abbreviations* ..... *Fehler! Textmarke nicht definiert.*

1.4 REFERENCES ..... **FEHLER! TEXTMARKE NICHT DEFINIERT.**

**2 SYSTEM APPLICATION DESCRIPTION .....6**

**3 PROJECT PLANNING .....7**

3.1 PLANNING OF MAIN ITEMS.....7

3.2 RISK MANAGEMENT.....7

3.3 TAILORING MEASURES.....7

3.4 ORGANIZATION DIAGRAM .....8

**4 PROJECT CONTROL.....10**

**5 CONFIGURATION MANAGEMENT.....11**

**6 TESTING .....12**

**7 QUALITY ASSURANCE .....13**

7.1 Q TRAINING .....14

**8 MISCELLANEOUS .....15**

8.1 DEVELOPMENT PLATFORM.....15

8.2 TRAINING (OPTIONAL).....15

8.3 PLANNING PROCEDURE (OPTIONAL, WHERE REGULATED ON PROJECT LEVEL) .....15

8.4 PROJECT CONTROL PROCEDURE (OPTIONAL, WHERE REGULATED ON PROJECT LEVEL).....16

**TABLE OF FIGURES**

ERROR! NO TABLE OF FIGURES ENTRIES FOUND.

# 1 Introduction

## 1.1 Purpose and Scope

State the purpose and aim of the document. If applicable, it may also be an abstract at this point, which summarizes to contents in a nutshell.

State for which project, department or group, etc. this document applies.

## 1.2 Definitions, Terminology and Abbreviations

Adapt and expand the below abbreviation table as necessary for your purpose. Further add in this section any definitions which the document is based on and which are necessary to make the contents clear.

Table of abbreviations used in this document

Abbreviation	Description
TBD	To be defined

## 1.3 References

/1/ Reference to any document quoted in this document

/2/ Reference to any document quoted in this document

/3/

## 2 System application description

This chapter shall provide the embedding of the software sub-project in the overall project by showing incorporation from

- Technical **and**
- Project schedule point of view (for project organization, see below).

Either

<Reference to the system specification (if available)> **and**

<Reference to the (system) project planning>

or

- Give more details, only if the system description does not exist or system definitions require additional information

Describe all necessary information from the system,

- System features  
It is recommended to describe even features decided not to be implemented in SW. This will allow tracking, whether additional workload has to be performed throughout the project life.
- List all delivery of samples (**not necessarily Software deliveries!**) viewed by the (system)/project in terms of functionality from the first to the last

For software variants, the decision for separate SW development plan documents should be based on criteria, like

- Same vs. different budget,
- Same vs. different project milestones,
- Related vs. de-coupled variant delivery contents,
- Groups of variants.

## 3 Project Planning

### 3.1 Planning of Main Items

**Reference to a detailed planning from a project planning tool (if available for all project members)**

In most cases it is appropriate to point to more than one plan/add some planning to a graphical resource planning.

The planning should at least contain

- **Estimates** of planning items (e.g. efforts, technical resources) **and assumptions** that led to it
- Esp. **schedule** (work packages and responsibilities) **and assumptions** that led to it
- Main software **milestones** (e.g. specification freeze, delivery dates, sample delivery, planned project milestones)
- A **list of** internal and external **commitments**, the software project has to deal with. (e.g. schedule commitments, dedicated form of customer documentation causing additional efforts)

This can be among others a plan with all deliveries, a plan with intended reviews, and all intended baselines/freezes.

These plans can be graphics integrated in this chapter, or simple tables, or documents of its own which can be added to the project folder.

### 3.2 Risk Management

This chapter must contain a list of

- **Identified** (description of possible risk),

Identification of software risks can be **supported by**

- **Examination of constraints** which can lead to deviations from the planning; this part should describe in detail everything that could limit/endanger the agreed time/cost/quality frame of the project,
- **General comments on the development approach**, which (non-)practices limit/endanger the agreed time/cost/quality frame of the project,
- **Identification of risks in case of failure** of the software (e.g. in form of a safety concept or by determination of the safety integrity level according to ISO/IEC 61508). This information may be a part of the software requirements specification documentation.
- **Assessed** (severity and probability of occurrence), **prioritized, and**,  
Additionally, information like
  - Time frame-to-occur,
  - Circumstances and events that make the identified risks more or less likely to occur, and
  - Preventive (i.e. prevent risks to occur) and corrective (i.e. reduce effects) measures may be documented to support risk tracking and risk management activities.
- *Decided risks, providing a (reference to the) risk action list*

### 3.3 Tailoring Measures

There are no tailoring measures necessary for the project.

In case you apply any of the allowed tailoring measures e.g. for sample phases, describe them here and delete the above sentence.

### 3.4 Organization Diagram

- The aim of this diagram is to fix responsibilities and make the participants of the software team aware of the different roles they take on during a software project.
- Responsibility description; name **all** team members of the software project
- For **roles different to the defined roles**: If there are
  - **Differences in semantics** of roles in SW development or
  - **Extended responsibilities** (e.g. SW team leader has dedicated responsibility for budget) or a
  - **Role empowerment** is practiced (e.g. a developer empowered as SW architect), **a detailed role definition should be appended** at the end of the table.
- A detailed **definition of interfaces, roles and responsibilities**, if
  - **Different phases** of algorithm development cycle are **performed by several groups** or organizational units (e.g. software requirements specification; a dedicated test team is existing; system test not performed by software department) or
  - Software **development is spread over different software groups** (e.g. dedicated groups for Concept, Implementation, Test, Calibration) or
  - Software development is fed by/delivering to **libraries**.
- For small projects: do not cross out lines but assign more than one responsibility to one person

Responsibility	Person		
(System) Project Manager (PM)	<name>	<department>	<phone number>
Software Project Manager (SW PM)	<name>	<department>	<phone number>
SW Change Control Board (SWCCB)	<name>	<department>	<phone number>
		•	
		•	
		•	
Software Configuration Manager (SW CM)	<name>	<department>	<phone number>
Quality Planning Engineer:	<name>	<department>	<phone number>
Software Quality Engineer (SW QE)	<name>	<department>	<phone number>
Developers (SW D):	<name>	<department>	<phone number>
		•	
		•	
		•	
Software Testers (SW T):	<name>	<department>	<phone number>
		•	
		•	
		•	
Customer Interface:	<name>	<department>	<phone number>
– Internal (for large projects also name interfaces to HW, generic/basic SW, library management etc.)		•	
– Customer side		•	
		•	

Either

All roles mentioned above are implemented according to their role description in the related procedure.

or

Detailed description of the extended or empowered role in SW development.

If development activities are spread over different groups a description of responsibilities and interfaces should be added here.

## 4 Project Control

### Comparison of actual results against planned schedule

- Give interval for check of actual project progress against schedule
- Name attendants and kind of meeting(s), e.g. SW team, core project team, (customer) coordination meetings

## 5 Configuration Management

The configuration management for the project is performed according to the standards and structures defined described in the "SoftwareConfigurationManagementPlan.doc".

## 6 Testing

The testing for the project is performed according to the standards defined in the "SoftwareTestPlan.doc".

The detailed schedules for the testing activities can be found in the general project schedule. For specific sample deliveries and algorithm releases the testing activities are tailored as follows:

<b>Delivery / release</b>	<b>Planned testing activity</b>	<b>Remarks</b>

## 7 Quality Assurance

This chapter constitutes a quality assurance plan for the software.

### 7.1 Review Plan

The following information has to be documented within the review plan: the objects to be reviewed, the review method, the date and the responsible and the recommended participating roles. All information should be defined project specific within the following table. The detailed planning on dates, responsible persons and participants are defined in the project plan of the project. The corresponding names to the roles are documented in the Organization Diagram in this document.

The Review Plan has to be consistent with the objects defined within the Process Description. See the table below: structure and review objects (to be completed regarding the Process Description).

Remarks:

- ⊙ Mandatory
- WT Walk Through (may be by Netmeeting)
- IN Inspection (offline check of results and documents followed by a meeting / Netmeeting to discuss the results).
- M Review Meeting

Phase	Review Object	Method	Responsible	Participants	Comments
ObjectR	Software Requirements Spec				
	Software Development Plan				
MilestoneR	S1				
ObjectR	Software Design Document				
	Source Code				
MilestoneR	S2				
ObjectR	Software Test Plan				
	Software Test Specifications				
MilestoneR	S3				

### 7.2 Quality Goals

Following metrics shall be used, which are reported in the CM Report of the project at the reporting date:

Goal	Metrics
<b>(G1) Find all errors before software delivery to customer</b>	(M1.1) Number of errors after delivery ( <b>EAD</b> )
	(M1.2) Review Fulfillment for Software Design ( <b>SW Design</b> )
	(M1.3) Software Testing – actual vs. planned ( <b>SW Testing</b> )
<b>(G2) Compliance to the software development process</b>	(M2.1) Milestone Reviews – actual vs. planned ( <b>RevM</b> )
	(M2.2) Software Planning – actual vs. planned ( <b>SW Dev.PI.</b> )
	(M2.3) Object reviews – actual vs. planned ( <b>RevO</b> )

<b>(G3) Stability of software requirements</b>	(M3.1) Stability of Requirements ( <b>SW Req.</b> )
--	---

The Software quality goals are:

	<b>S1</b>	<b>S2</b>	<b>S3</b>
<b>SW Req.</b>	100,0%	100,0%	100,0%
<b>SW Dev.PI.</b>	100,0%	100,0%	100,0%
<b>SW Design</b>	60,0%	100,0%	100,0%
<b>RevO</b>	28,0%	42,0%	100,0%
<b>RevM</b>	25,0%	50,0%	100,0%
<b>SW Testing</b>	0,0%	0,0%	100,0%
<b>EAD</b>			0,0%

### 7.3 Q Training

No special quality trainings are required for the project. The staff shall participate in the standard quality and process trainings as defined and offered by the organization.

Definition respective reference to training concept & plan

- Topics of training: QA basics / overview, QA process, review methods, tools, etc.
- Roles to be trained: SW-QE, PM, SW-PM, SW developer, Calibration engineer, management

## 8 Miscellaneous

Add additional subchapters, if appropriate.

### 8.1 Development Platform

Describe

- All implemented means to develop the software (i.e. development platform incl. a list of all tools with version) and
- All constraints that will be associated with it (software and hardware level).

### 8.2 Training (optional)

Derive (at least at the project start) and document all training needs (standard and special) necessary to ensure the

- Technical (e.g. development platform, micro-controller),
- Administrative (e.g. techniques for project management) and
- Organizational (e.g. environment regulations for development)
- skills of the SW-PM, the SW developers and the CM responsible.

It should regard the needs of the project to accomplish the project successfully.

*It can be performed by e.g. providing a **Reference to BF standard training program** and/or*

- A list of training needs for skills mentioned above.

### 8.3 Planning Procedure (optional, where regulated on project level)

It may be useful on project level to make the procedure to derive plans visible and repeatable, especially if no organization wide regulation exists.

The following topics should give an idea which topics should be covered by a planning procedure.

The documentation should contain

- A description of the work breakdown structure (i.e. division of the software project into work packages representing individual work units that can be separately assigned, performed, and tracked. This should form the framework to plan and control the work done on the project).
- An overview how long-term planning (e.g. via an effort estimation meeting) and short-term planning on a more detailed level are performed.
- An estimation procedure how to derive estimates on software size, effort and costs, ECU resources and development facilities.
- A procedure how to derive schedule planning.  
The procedure deriving plans should cover stages, such as
  - Structural planning
  - Size and effort estimation planning
  - Sequence planning (of work packages) and
  - Scheduling.
- An identification of planning items including
  - Size estimates (incl. definition of the granularity to decompose SW products),
  - Effort and cost estimates (mirroring the embedding development processes, skills of staff and managers, the capabilities of tools used in development environment and the development facilities needed and allowing to finally estimate total effort and cost for single deliveries and the project),

- Estimates of critical computer resources in the project products (identifying and estimating them, e.g. memory capacity, processor power, communications channel capacity, peripheral capacity),
- Plans for development facilities (defining the plan for obtaining the facilities, assigning responsibilities and negotiating commitments),
- The software development's schedule (basing on preceding stages and availability of staff, resources and facilities, covering imposed milestone dates and critical dependency dates (on software development and project level), determining time phasing of activities), and
- Internal and external commitments (negotiation, agreement, documentation)

as well as an instruction how to describe the assumptions which form the basis for the respective estimates.

- The approach, the mechanisms and the intervals of bringing changes from software development (schedule) planning into overall system project planning and vice-versa.

Note that the planning and re-planning data are the basis for control. They should allow determining both progress in real project life and the suitability of current implementation of single process stages.

#### **8.4 Project Control Procedure (optional, where regulated on project level)**

It may be useful on project level to make the procedure to perform project tracking visible and repeatable especially if no organization wide regulation exists.

Therefore the following topics should give an idea which topics should be covered by a project tracking procedure.

The documentation should contain

- A description of the mechanisms to determine the actual values of the planning items, because they form the basis for tracking.
- A description of the mechanisms to track, control and manage the planning items, especially
  - Made assumptions,
  - Size,
  - Resources,
  - Schedule,
  - Identified risks and
  - Internal and external commitments.
- A procedure of handling serious deviations and the according escalation mechanisms that should be followed.