SDEVEN Software Development & Engineering Methodology

Version: 7.0.8
Release date: 230730

Blueprint. Processes & Deliverables (SDEVEN.90-RENBLU)

Table of Content

Preamble

This section is about whole methodology **phases**, **processes**, **steps**, their *codes*, all over a cheat sheet / blueprint of "the whole".

The phases are important as **codes** (on *documents* for example) and as name and content for a better understanding of process. They can also be used (and this is recommended) *as directory / file names* for a common understanding (at least *830-DEV* and *880-RLSE*) but when use them in file system (file or directory names) is important to pay attention to case (**lower ase is accepted**) and the character `-` (dash) and space (**underscore `_` is accepted**).

This methodology, being applicable to all software development projects, not all sections are applicable in all cases, so in design template document - md will be CLEAR specified for each section if it is REQUIRED or OPTIONAL.

Blueprint / typologies

Basically the **RENware SDEVEN)** (aka *SDEVEN*) methodology name is an acronym of *Software Development & ENgineering* and is adapted to company *practices*, *software styles*, types of software in portfolio and scope and follows a classic *waterfall* inspired pattern but which can (and is) be used as an evolutionary model with sprints inspired from Agile methodology.

There are classic phases up to release a product followed by maintenance and current operations phases post release. After a release, the methodology allows returns to any of the first 6 phases for new features / versions releases, the MAIN OBJECTIVE being a higher coherency, traceability and ability to move project from a team to another team with minimum overheard due to project adoption.

In this document processes deliverables are summary described keeping a level that facilitate the understanding of principles of methodology. For more details is recommended to consult the appendix referred (see references section).

Roles & responsibilities

For each "final" (ie, leaf) process, the basic required roles was specified. Roles are abbreviated as described in *10.ADM* section. Also a super generic role was used in places where more and different specific roles should be used depending on project context, size and complexity.

Phase 100-ANA Analysis

This phase is the stage where the system requirements are collected and processed (engineered) to a more structured form in order to be more usable in system / product elaboration process.

Traditionally the phase start from system requirements which are (re)structured in primary designs (aka high level design) at a level which allows for system blueprint, objectives, structure and functions.

Also, here are produced the sales kits which will be used by marketing and sales structures in their work to market and sale the system / product.

110-SRE System Requirements

These are the requirements as collected (elicitation) and categorized by relevant taxonomies. As mandatory taxonomies are:

- **functional** types: functional, implementation issues, performance issues, user interaction issues, interfacing issues, standards compliance
- **urgency and importance** types following MoSCoW paradigm - must, should, could, would

Roles & Responsibilities

- REQUIRED: ban
- RECOMMENDED: rad, twr
- OPTIONAL: sen

120-CPTS System Concepts

This is what is called usually *High Level Design* but it means more than it. This deliverable should contain the following:

- the basic system concepts as business specific terminology
- a high level architecture indicating at least the basic functional components of the system (logical architecture)
- the system universe as other business components which interact with the system and very shortly what kind of information is exchanged in interaction process
- what are the applicable standards and why are they important
- what are the main security issues and how can be avoided
- types / models of user interfaces intended to be implemented
- which logical components are back ends, front ends, communication, interfaces

Roles & Responsibilities

- REQUIRED: ban, sen
- RECOMMENDED: rad, twr
- OPTIONAL: prm

130-SKIT Sales Kit(s)

The "things" required to sales structures should be put under this code, eventually with detailed documents and codes following a class notation (with dot character). The most common things are:

- a system blueprint that can be used in presentations

- a system logical architecture that can be used in presentations
- some key differentiators that can be used in offers and bids
- various pro/con-s analysis ref other market similar products
- a list of system requirements that it satisfy (and usual responses) usable in bidding process
- a list of system standards that it is complaint to (and usual responses) usable in bidding process
- some known "frequently asked questions" and the answers to them

None of these represent new information. All information os to be found in the other documents just it is (re)structured in other way in which is more useful to sales personnel.

Roles & Responsibilities

- REQUIRED: sen, prm
- RECOMMENDED: twr
- OPTIONAL: pm, sect

190-SKTD Sketches & Technical Diagrams

These are different drawings of system parts. They are (should be) separated because can be useful (as raw images) in other places / processes such as presentations, UI design, branding, etc.

Phase 800-SWD Software Development

This phase covers the software realization process but from an engineering point of view, meaning not only code and programs writing, but also the design, testing and release processes, Aim is to make a **SYSTEM PRODUCT**, usable and maintainable both by the beneficiary and RENware as the producer. Another important issue is related to capability of the system to be "migrated" to other team members with as less overhead effort as possible.

810-DSGN System Design

This is the main technical detailed document of the system. It is the starting point for any developer that join the project snd its job require to understand the system.

Roles & Responsibilities

- REQUIRED: sen
- RECOMMENDED: twr, sect
- OPTIONAL: prm, ptm

820-SYINT System Internals

This is nothing else that low level design too, but is seen as a *dedicated chapter or appendix* of 810-DSGN and refers technical documents (that often are read singularly, for example a dev needs to review only some states of an object) such as system states, sequence diagrams, etc, documents that by default are classified and should not get out of project without special approvals.

Roles & Responsibilities

- REQUIRED: sen
- RECOMMENDED: twr, ptm
- OPTIONAL: dev, ban

830-DEV System Development

This "contains / is" the system as it is on repository system. Practical is the local repository image. It clearly contains all things / codes / programs / scripts needed for system to work, bur it can contain (and it is strongly recommended) the system other documentation being it technical or end user manuals.

The only thing that must be said is regarding documentation. Therefore the *technical one must be in text formats* (markdown is preferred) and *other documents* should be also in *text format as much as possible*. Binary formats should be strongly avoided (are not "`diff`-able") but in "no other option" cases, the repository administrator must be informed about binary files to action in consequence. As other suggested options to binary documents would be:

- use markdown extensions like `mermaid` for "graphical" things / diagrams,
- store them as xml format,
- store them as pdf format,
- use "open text formats", etc.

Roles & Responsibilities

- all kind of required `dev`
- sometimes would be necessary `ptm` for different clarifications

840-TEST System Testing

The things that are mandatory here are:

- a **testing plan**: what to test (*REQUIRED*), when to tets, by who
- **test cases** - that represent for each test required:
- the exact expected steps to be done with a number / code to be easily referred in other places

- the expected result to be obtained by doing previous steps (often is useful to indicated an acceptable tolerance, especially when are talking about numbers or times)
- the system reported messages (if there are) and results, preferably with associated print screens
- a **testing report** that summarize the results of testing process: number of severe / critical bugs, number of acceptable errors (as solvable by work around), recommendations to improve, documentation non conformities, time spent for testing, automation tools used, required and recommended.

Roles & Responsibilities

- REQUIRED: func, scat, sect
- RECOMMENDED: twr,ptm
- OPTIONAL: isat

880-RLSE System Releases

This refers to releasing process regardless of released version qualifier (not only to *release versions*) but to any kind of version being it an alpha or a beta or a previewer, etc.

Any release should pass a test process which is done by developer itself as first test (**alpha test**). No promote to higher version qualifiers is allowed if alpha tests did not pass or was bypassed (skipped).

After alpha test can be created a branch for testers team or a tag (that will be used latter for a test branch creation).

Any tag will be kept at least for minimum 4 tags after it but no more than 20 tags after. These min and max can be modified if project require other limits.

Tags will be saved also on an external backup (in project history).

Releasing a version must be documented accordingly in project *CHANGELOG* and if project require, a *release note* should be issued. Both must be written with enough care to text explanations and having in mind that will be used latter by other persons in writing different product documents and marketing materials.

Roles & Responsibilities

- REQUIRED: isat, scat, sect
- RECOMMENDED: prm, ptm, twr
- OPTIONAL: pm, dev

890-MNT System Maintenance

This phase is about:

- Maintenance Plan

- Hot Fixes
- Critical Patches
- Other Updates

all of these mainly in relation with "beneficiary" os system / product / application.

Roles & Responsibilities

- REQUIRED: sen, ptm
- RECOMMENDED: sect, twr
- OPTIONAL: pm, dadm

Phase 900-OPS Operations

This phase is about maintaining the system after a release. In fact is about ***maintaining a release***.

Few words about software maintenance

The maintenance is probably the hard thing about a software. Software maintainability is the most important issue for any software company that develop its own software as portfolio product.

Along decades there was more and more improvements in software development process, probably the biggest step being made when *software development was considered an ENGINEERING discipline* with all things that engineering means. But the biggest issue remained the *measurement process*.

From maintenance point of view a released version must be kept "alive" a period of time. That alive is mostly generated by the fact that software being an intangible product, so hard to measure, always could be some astigmatic functionalities, ie non conform ones, aka bugs, that should be remediate after release. Different methods to categorize and measure these non conformities was developed in time. **Now the most accepted "definition" for a "bug" is: a(ny) thing that works otherwise that is written in product documentation**.

910-MNT System Maintenance

These are current operations related to modifications /changes that must be made to a released version.

The important things are:

- to maintain an acceptable coherence during modifications in order to avoid ***side effects***
- to make changes in a way to not be repeated after some time, ie to not make / solve the same issue more than once in time!
- to present an interface to users that is able to collect and centralize issues (aka support line / center)

Roles & Responsibilities

- REQUIRED: dadm, radm, ptm
- RECOMMENDED: sect, twr
- OPTIONAL: pm, prm

920-TLE Prepare temporary live environments

This activity refers to dev infrastructure administration and in fact, from software development methodology point of view means that who requested for this environment should create a kind of checklist with requirements in order to be easily verified by administrator.

Roles & Responsibilities

- REQUIRED: dadm
- RECOMMENDED: ptm
- OPTIONAL: dev

990-PMSP Project Management Support (REQUIRED)

This activity refers to technical support activities required in Project Management, ie estimations, deliveries breakdown and quality factors in PoC calculation, etc.

Roles & Responsibilities

- REQUIRED: sen, prm
- RECOMMENDED: ptm, pm
- OPTIONAL: ban

References

A comprehensive template as suggested structure and content can be found here - format md.

Last update: July 30, 2023