SDEVEN Software Development & Engineering Methodology

Version: 7.0.8
Release date: 230730

Administrative policies (SDEVEN.10-ADM)

Table of Content

Preamble

This section describes the organization environment and framework of software development structure.

Staffing roles and responsibilities

Testing

- **funt** *functional tester* - test the system from the functional point of view (black-box testing)
- **scat** *external standards compliance tester / auditor* - check the system declared external standards for compliance at least minimum mandatory requirements; a;so check if standard is still active, used at least as best practices and relevant for system

- **isat** *internal standards tester / auditor* - check if the system is compliant with company applicable and relevant standards
- **sect** *security tester* - check for system security according to usual practices, known / usual attacks and vulnerabilities

DevOps and systems administration

- **radm** *repository admin* - assures the project / software repository(es) are up to date, clean and can be accessed by team
- **dadm** *dev infra admin* - assure the drvelopment infrastructure / environment for a project, meaning at least: development, testing, production like, live demo machines, their installation, cloning, backup, making iso images, availability (from different locations according to needs), security issues, and so on

Project management

- **prm** *product manager* - assure the system is developed according to roadmap and company strategy; check that a version from roadmap is completely and well defined and is relevant for market and as current best practices; checks financial aspects of projects
- **ptm** *project technical compliance tester / auditor* - check if a version that is intended to be released is complete enough to be released (documentation, migration, and so on)
- **pm** *project manager*

Development and research

- **dev** *developers* - assure code, programs, scripts etc writing
- **sen** *software engineers* - assure technical organization, design, architectures, toolstacks, practices, patterns
- **ban** *analysts* - assure understanding of targeted information domains, elaborate functional testing strategy plans
- **rad** *researchers* - assure discovery and usage strategies for technical market (best) practices and patterns; elaborate methods for different technologies usage, elaborate integration strategies
- **twr** *technical writer* - write and check technical documentation

Projects and teams organization

Project aspects

A project is officially started through a management decision. This could be an informal one, but for a *clear* team allocation, budget, stuff nomination, etc, a written document published internally is preferred.

Basically projects can be normal / **standard** ones (ie, with standard classification rules) or **classified** with more strict rules regarding access to their information.

Other normal taxonomy establish projects as **internal** (for internal company use or for research) or **external** for those being a target beneficiary of project results.

Team aspects

Teams are dynamically allocated per projects as needed. (**DYNAMIC ALLOCATION**) Any member allocated a time frame on a project will follow (in that time frame) the specific project rules and organization. (**STATIC ALLOCATION**) Out of allocation time frame will follow the fixed / administrative pattern.

Applicable procedures

For a detailed description of project management policies refer the applicable *Project Management Methodology*.

Working technical environments

Working environments can be classified as:

- *development* - usually on personal computer, but could be situations where one or more development servers are needed especially for remote work, operating system issues, processing power, testing on more "real" machines, etc
- *test* (aka QA-envs) - for testing issues, regardless by which members are (to be) done
- *production* systems - also for testing issues but used in final stages, just before committing work to client users. These environments should be as much as possible very appropriate to a real machine that exists in current use at client

Escalation procedures and practices

From this perspective a software project should be seen like any other project. Therefore is anything special than normal procedures used in *Project Management Methodology*.

Last update: July 30, 2023