Contents

[1. Design method 1](#_Toc502130181)

[1.1. Preliminary design 1](#_Toc502130182)

[1.1.1. Preliminary software design 1](#_Toc502130183)

[1.1.2. Dataflow design 2](#_Toc502130184)

[1.1.3. Preliminary design of the production environment 2](#_Toc502130185)

[1.1.4. Detailed design of the test environment 2](#_Toc502130186)

[1.1.5. Detailed design of the development environment 2](#_Toc502130187)

[1.2. Detailed design, development & testing 3](#_Toc502130188)

# Design method

The design method describes the steps needed for project start-up and the transition from these startup steps to a structured work method for the project.

## Preliminary design

The design of the disclosure system will be completed in iterations, where each iteration will be termed a work cycle. A work cycle is a two weak block of work encompassing detailed design, development and unit test of a set of functions defined at the beginning of the cycle. Before the cycles can begin the following preliminary design elements need to be created.

* Preliminary software design
* Dataflow design
* Detailed development infrastructure design
* Detailed design of the physical development environment
* Detailed design of the test environment
* Preliminary design of the production environment
* Preliminary project plan

### Preliminary software design

The preliminary software design defines the scope of the project and also breaks the project into manageable blocks. These blocks should be independent pieces of software with such well-defined boundaries that at least in theory can be developed independently of one another. Such a piece of software is defined as a module.

A preliminary design is made for each module consisting of the following components.

* Major functions and UI elements are listed. Opportunities for creating reusable components that can be shared across modules should be evaluated.
* For each function an evaluation is made highlighting whether there are specific legal requirements impacting the design of the function. Legal requirements arise primarily from TSYRT 2016:46 and 2016:47.
* A risk analysis is performed for each module. The risk analysis should encompass the potential for the loss, manipulation of or unauthorized proliferation of data though malfunctions, external factors and malicious intent.
* Finally preliminary test scenario is also defined. The test scenario is not to be seen as a specific unit test but should be seen as a minimum criterion for completion of the module.

### Dataflow design

When the modules are designed a preliminary data flow diagram is made describing the dataflow and how the modules cooperate to create a process. The dataflow diagram should visualize how the system will work in practice upon completion.

### Preliminary design of the production environment

Preliminary design of the production environment will outline all components of the production environment such that it in theory could be built. It is still termed preliminary as depending on the results of stress testing the systems the platform may need to be scaled up or down. The production design set a number of prerequisites for the test environment which for example should be dimensioned to have lower performance machines. The preliminary production design will also include a risk analysis.

### Detailed design of the test environment

The design of the test environment can easily be done after the production environment has been designed. Operating systems should be identical to the production environment and hardware specifications should be lower than the production environment. The test environment is to be separated from the development environment according to 2017:47 ch.2§4. To conform to this requirement of separation the test environment shall use dedicated machines and be run on an isolated LAN with no Internet connection. Tests will be performed using executables only on the test environment. Development tools may not be installed on the test environment to ensure that installation procedures are correct and complete. Development tools may install software that the executables are dependent without the developer being aware of this. Unit test may be carried out in the development environment whereas module testing will be carried out in the test environment.

### Detailed design of the development environment

The development infrastructure consists of the software & hardware tools necessary to complete the project. This encompasses the coding languages that will be used, the selected development environment for these languages and third party tools such as software libraries, database management systems etc. The detailed development infrastructure design will also include a risk analysis for the development project covering loss or manipulation of data, unauthorized access & proliferation. Developers may not install or use any software other than that specified in the development infrastructure design for work on the disclosure project.

### Preliminary project plan

The last phase of the preliminary design is to make a preliminary project plan. The preliminary project plan has only 2 purposes.

* To provide a “guesstimate” of the project duration
* To provide the order in which the modules need to be developed.

To rate the relative difficulty and size modules according to the scales (Small, Medium, Large) for size and (Simple, Average, Difficult)

Having rated the modules, the project team decides how long it takes to design, develop and test a small and simple module in man days. This is the base unit of (B.) The table below will then be used to provide estimates for all other modules.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Small (x1) | Medium (x3) | Large (x5) |
| Simple (x1) | B x 1 | B x 3 | B x 5 |
| Average (x3) | B x 3 | B x 9 | B x 15 |
| Difficult (x5) | B x 5 | B x 15 | B x 25 |

Deciding the order of the modules is a question of identifying dependencies between the modules a client will be dependent on a server and a server is dependent on a database etc.

When dependencies have been identified the modules are simply ordered.

## Detailed design, development & testing

With the preliminary design completed there in enough information to start the production phase of the project which, as already mentioned, will be conducted as a series if work cycles initially set to two weeks. A work cycle mainly consists of the detailed design, development and test of a number of functions. However any tasks (such as replace the broken project printer) that are project work should be managed in the work cycles.

To begin the work cycle the first module from preliminary design is selected for detailed DDT. The reason only one module is developed at a time is that when the module is completed the project team will have gained substantial insight into many aspects of the project work. It is important that this insights or lessons learned are captured early on and included in the DDT of all future modules. This will provide higher quality and shorter development times.

### Working procedures for each cycle

Each cycle is initially a 2 week period although this will change if circumstances so dictate. Day one of week one commences with a planning meeting.

The planning meeting will decide

* What functions are in need of development in the near future for the current module?
* What specific functions should be prioritized for completion in the coming cycle?

The planning meeting should have a meeting protocol based on template 1.4 Cycle planning meeting template.

The functions that are scheduled for the cycle should be completed in the cycle. A cycle should not leave unfinished work. It is better in the beginning to put too little in a cycle when there is uncertainty as to what can be completed. As more and more work is completed and uncertainty diminishes the amount of work in a cycle should be continually increased.

Completion of a function is defined by a finished detailed design, development, successfully completed test cases and any supporting documentation specified by the design such as procedures or user documentation.

At the end of each week a new version of system code, test code and project documentation should be uploaded to GitHub. Daily backups are made to the test file server.

At the end of the cycle is an evaluation meeting is held covering what is completed, what is not and why not. Impediments to timely delivery should if possible become actions items (the action being removing the impediment from future cycles) and the action item included in the next cycle if possible. The meeting should have a protocol using the template 1.5 Cycle Evaluation Meeting Template.

### Detailed Design

The first step in detailed design is to cross reference the functionality to be designed with the system requirements in TSYRT 2016:47 & TSYRT 2017:46 and assess what impact these requirements will have on the design of the function. Begin by summarizing these requirements in writing.

With the requirements in mind a list of high level functions of the functionality block. Even in detail design there is no point in listing every possible supporting function that will be needed or designing the exact flow of execution. This is pointless, as in it is only possible in theory. In practice, too much changes in the details for a too comprehensive design to be of use. Functions included in the detailed design should be:

* Utilities that are clearly a prerequisite for the module to work
* Functionality that a user should reasonably expect to be in place
* Any functionality explicitly required by the legislation that is a basis

The detailed documentation of the code itself is the code and making it readable. This is facilitated by following the coding standards and supplemented by the coding diary.

After the functions the major UI components of the module should be designed if any exist. A simple hand drawn sketch is sufficient for this purpose. Each sketch should have a written description to explain basic workflow and to link the UI elements to the list of functions.

During the design special consideration is to be given to whether existing reusable code items can be used or if any of the functionality should be moved out to an external library to assist reuse in the future. A section of the design covering reuse considerations should always be included even if no reuse opportunities have been identified.

The detailed design must also include a risk analysis for the function.

### Development

When the design is finished development can begin starting with reusable components. Development should adhere as closely as possible to the coding standards. As development progresses the design need not be updated unless a major hurdle is encountered that negates the existing design in which case the detailed design needs to be redone.

Each code function should be tagged with a unique identifier consisting of three numbers separated by a full stop. The first two numbers are the document numbers in the DMS the third number is a sequentially increased number to uniquely identify the code function in the coding diary.

During the development the coding diary should be updated at the discretion of the developer. Not every code function need have an entry in the coding diary. For detailed descriptions on what should be in the coding diary see the coding standards document.

### Testing

For each code function a unit test should be created. The test case should be a code project that can simply be run to execute the test.

A test case should fulfill the following requirements

* The test can only be executed in one way by running the test project. No other preparation should be necessary.
* Tests only one thing.
* Has only one result, if the test is passed true / false

## Module testing