**Software Selection Process  
Needs Initiation  
Modern Prototyping Software**

**Purpose:**

This document is the first one done by the Requester to initiate a request to purchase a new software The purpose of this document is to provide a description of the needs (business), to propose new software (LSA) and to assess the request (CRM).

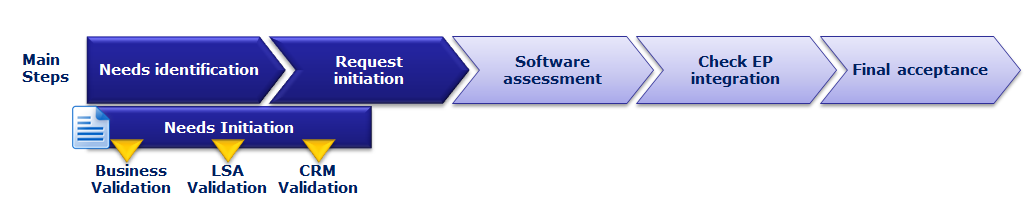


Table of contents

[1. Business – Software Request 3](#_Toc462057025)

[1.1. Summary 3](#_Toc462057026)

[1.2. Key target users 3](#_Toc462057027)

[1.3. Current situation 4](#_Toc462057028)

[1.3.1. Overview 4](#_Toc462057029)

[1.3.2. Description of weaknesses, risks in or drawbacks of the current system 5](#_Toc462057030)

[1.4. Description of needs 6](#_Toc462057031)

[1.4.1. Overview 6](#_Toc462057032)

[1.4.2. Justification 6](#_Toc462057033)

[1.4.2.1. Main reasons justifying the initiative 6](#_Toc462057034)

[1.4.2.2. Expected business benefits 6](#_Toc462057035)

[1.4.3. Description of Needs/Features 7](#_Toc462057036)

[1.5. Assumptions and Constraints 8](#_Toc462057037)

[1.5.1. Assumptions 8](#_Toc462057038)

[1.5.2. Constraints 8](#_Toc462057039)

[2. LSA - Software Proposals 10](#_Toc462057040)

[2.1. Software #1 10](#_Toc462057041)

[2.2. Software #2 10](#_Toc462057042)

[3. CRM – Software Assessment 12](#_Toc462057043)

[3.1. CRM Cross-check 12](#_Toc462057044)

[3.2. EP software list check 12](#_Toc462057045)

[3.3. Qualification 12](#_Toc462057046)

[4. Annex - Document control 13](#_Toc462057047)

[4.1. Validation 13](#_Toc462057048)

[4.2. Circulation 13](#_Toc462057049)

[4.3. \*in the SSPTRACK JIRA request 13](#_Toc462057050)

[4.4. Change history 13](#_Toc462057051)

[4.5. Applicable documents 14](#_Toc462057052)

[4.6. Reference documents 14](#_Toc462057053)

[4.7. Glossary 14](#_Toc462057054)

[4.8. Usage conventions 15](#_Toc462057055)

# Business – Software Request

Business Case applicable (IT Plan):

Clarity ID:

| **Document name** | **Document** |
| --- | --- |
|  |  |

If NO then FILL IN this chapter.

## Summary

The aim of this section is to explain shortly why and what kind of needs are expected.

Below is a list of features that are expected of a modern mockup making software that would relate to current UX conventions of a Single Page Applications that European Parliaments is producing.

* **Dynamic mock-ups** - with ability to mock major user-system interactions as a part of a single screen – text boxes, drop downs, tick & check boxes, basic form validation (most of what can be achieved with simple front-end JS code).
* Allow **reuse of component**s across screens, with changes to a component propagating throughout the screens using the component.
* No software installation operation – preferably a web application that would also be a test/presentation delivery platform.
* Produce **user testable** (supervised tests) pieces of UI.
* Preferably produce **HTML output**, also export to PNG and PDFs.
* Allow **reuse of common UI conventions** – web fonts, icons, frameworks Bootstrap/Foundation/Semantic UI or others.
* Allow **creation of own UI conventions** with aim of reuse at various screens & projects.
* Allow **comments from various users** (other team members or end users).
* Allow to edit the mockups by other team members.
* Gracefully handle iterations across individual screens and the multiple screen shared objects.

## Key target users

| **Category** | **Average number of users** | **Key beneficiary (Yes/No)** | **Number of concurrent access** |
| --- | --- | --- | --- |
| MEPs, Assistants, Political Groups |  |  |  |
| EU Citizens, Other EU Institutions, National Parliaments |  |  |  |
| EP Staff (User testing) | +/- 30 | Yes | 2 |
| IT Specialist (Technical specialist, Analyst tester, Developer, etc.) | 2-3 | Yes | 2-3 |
| Other (describe) – |  |  |  |

## Current situation

### Overview

This section identifies and/or describes the current set-up in order to provide a clearer picture of what is required.

For the last few years a clear new design pattern has been established in the IT industry – a Single (or limited) Page Products. There has been a progressive transition that started in 2005 with publication of [Ajax: A new Approach to Web Applications](https://web.archive.org/web/20080702075113/http:/www.adaptivepath.com/ideas/essays/archives/000385.php). In 2006 W3C has released the first draft of XMLHttpRequest Object and the technique entered into an Official Web Standard antechamber.

The technique has evolved since mid 2000’s and along the way solved a number of usability and client responsiveness issues in IT products. The European Parliament, for some applications, is also applying to the Single Page (or few pages) Applications – AT4AM, DST, AT4LEX. All these are complex, highly transactional, web product that hide client-server responses behind a persistence mimicking UI layer.

However, though the technique has provided with a more stable native application-like front end, it has introduced a host of new issues.

The problem that affects the authors of this request is lack of early mock-up and design tools that could be used to mimic complex and dynamic UI of the Single Page Applications. The current EP standard is WireFrameSketcher – a very eloquent tool to build pre 2005 - form-based, server side generated pages.

With WireFrameSketcher any attempt to build mockup of a similar (or the same) screen with various states, leads to multitude of separate screens. So far, so good, the issues start surfacing at the following moments:

- Attempt to change a single element shared across these pages leads to an inevitable frustration once some of an element requires updating (based on design result or with user feedback). Now, a designer must manually propagate these changes across all the affected screens. There are no independent design object that could be nested in screens, where changes to the object is reflected on all the screens that use said object. All changes must be manual, and every screen has no inheritance from any other screen. This makes mock-up design in WireFrameSketcher a very laborious process.

- Testing the screens with users leads to another problem, which may not be obvious before. As a single page now handles host of logic the same logic must be transformed across all the multiplied pages. Once any logic changes is introduced, it must be introduced in all the remaining pages that are to represent the one single page of the designed system.

The amount of labour required for frequent updates – that are at the very centre of User Tested – early prototyped designs - leads to a subpar work as a designer/analyst optimises the process for time. It would likely result in analyst attempting to mock up only one or few of the selected states per complex Single Page Application. End result of WireFrameSketcher is a limited mockup that does not present any (or little) of the dynamism that will be part of the final front-end design.

- Another of the points touched in this document is one layer of fidelity that WireFrameSketcher provides. This should not come as a surprise, given the name of the product. After all it is to produce sketches of UI with wire frames. Unfortunately this approach while great at the beginning of the design process, must be gradually elevated as the set of functionalities are being established.

- Lack of common UI patterns that come with a set of de-facto industry standard CSS front-end frameworks – Bootstrap, Foundation, Semantic UI and more. All these solve the same problem in a very similar ways. The biggest advantage these provide are commonly recognised UI conventions.

### Description of weaknesses, risks in or drawbacks of the current system

This section sets out the current weaknesses, problems or risks which prompted the initial needs study. An order of priority can be set, based on how serious an adverse impact the various problems have on the current system, as follows:

* 1: serious adverse impact, to be dealt with as a priority;
* 2: adverse impact - the problem can be got round, but should be solved in the new system;
* 3: no adverse impact - user-friendliness issue which can be dealt with in a later version but needs to be highlighted.

| **Problem** | **Description** | **Priority** |
| --- | --- | --- |
| **Non dynamic mockups** | Currently supported software produces only PNG exports that do not contain any interactions. There is no software that would allow to (at minimum) connect certain fields of a given PNG export to another PNG screen. | 1 |
| **No shared elements across screens** | Any element on any given screen is an independent design, there is no inheritance that would allow to propagate a change in one major element into all screens that inherit the element. Any modification must be propagated across all the screens by hand – and given that no screen contains dynamic elements, the number of screens requiring updates over a relatively minor change mounts up very fast. | 1 |
| **Remains at the level of wireframes – does not support other fidelity levels.** | Once a basic functionality has been established, there is no possibility to produce test screens with an intermediate fidelity – as to \*not\* break users focus on functionalities. Users tend to focus on presentation if the presentation is provided at a too low of a fidelity for a test. | 1 |
| **No HTML code to speed up start of the development** | Exports are provided only in raster files – PNGs/JPGs. Since the screen construction are simple wireframe primitives, the export cannot be pushed to HTML. However, if the screen mockups had been done with HTML primitives, the export could – in principle – be a HTML files. | 2 |
| **No CSS code to speed up the start of development** | As above. | 2 |
| **Does not mimic target (web browser) environment for reliable user testing.** | Lack of dynamism in the screens, raster graphics no HTML & CSS, result in deliverables that are more appropriate for a slide show then a reliable User Test. Currently – without an additional piece of software to map exported raster files to each other, there is no way to design a User Test path, where a mockup would respond to user action depending on area of the mockup that user clicks on.  Since these are just raster files, the entire User Test is missing a basic illusion of a software UI environment – one where elements respond to mouse and keyboard actions (e.g. how can a user type a message inside a PNG file?) | 2 |

**Priority**: 1: High; 2: Medium; 3: Low.

## Description of needs

### Overview

The aim of this section is to provide a clear picture of user needs and expectations. Needs are conditions that the software must satisfy. The first step is to identify, forward and record the basic requirements that the initiative must meet. The needs must be described clearly and simply, so as to ensure they are understood by all the SSP stakeholders.

### Justification

### Main reasons justifying the initiative

In this section, you have to justify why it is important to launch an initiative, you can also resume the main current weaknesses, problems or risks that can affect the organisation.

### Expected business benefits

In this section, you have to describe the business benefits, like:

* **Efficiency**: Will the requested activity/application contribute to increasing work efficiency/productivity in your service(s) and/or others?
* **Optimise resource**: Will the requested activity/application contributes to optimising the use of non-staff resources (e.g. technical and non-technical infrastructure resources, information resources, etc.) in your service(s) and/or others
* **Enhance capability**: Will the requested activity/application contribute to enhance management and delivery capability in your service(s) and/or others?
* **Improve service**: Will the requested activity/application contribute to improving service(s) delivered to users, either internally or externally?
* Etc.

Ability to build fit-for-purpose mockups, that would serve as a basis for User Testing sessions, not only as a paste-ins to Specification Document.

Ability to produce an interactive system prototype of varying fidelity is fundamental in solving two major design challenges:

* How adequate the system is to the problem that is being solved,
* User buy-in for changes the system introduces.

With a quality prototyping framework, that would take advantage of major existing front-end frameworks and allowing to reuse project/service/unit/directoriate/organizational level standard components would not only speed up the design process, it would also achieve the following:

* Better UX design as the major front-end frameworks provide a refined usability of UI primitives (primary vs secondary buttons).
* Faster front-end development, as the common components are reused,
* Better interaction with a UI designer (if one is available to the project team) – if there is no UI designer in the project, the common frameworks’ UI components would help that major UI/UX mistakes are avoided.

A good prototyping tool considers how its products will be used in interactions with End Users – and attempts to provide a decent testing environment in the target environment of the system. In our case (and most of other EP applications) the target environment is a web browser.

### Description of Needs/Features

This section sets out the main features required in order to meet the needs expressed, on the basis of the description given in the previous section.

| **Need/Feature** | | **Description** | **MoSCoW**  **Priority\*** |
| --- | --- | --- | --- |
| 1 | Dynamic mockups | It is possible to explain the feature through the user story concept: **As a** XXXX **I want to** XXXX **So that** XXXX  *Example:*  **As a** meeting organiser **I want to** print the final participants list **so that** I can check the presence  As a System Designer I want to be able to include a set of basic interactions in one screen (e.g. I want to be able to include text boxes, drop downs), so that I do not need to create a separate screens for each state (e.g. one for closed, one for opened dropdown). | M |
| 2 | Reuse of common components | As a System Designer I want to be able to set and reuse common components with their basic behaviour (e.g. button highlight and cursor change on mouseOver), so that I can build mockups that can be user tested. | M |
| 3 | Ability to start test sessions in the application’s target environment (web browser) | As a System Designer I want to be able to prepare and start a test/presentation session right in a web browser, to try to mimic the target environment that users will be taking. Keeping target environment is essential for decent User Test, as the UI of the browser takes part in the overall system design. | M |
| 4 | UI Object inheritance | As a System Designer I want to take any basic object, customize it for the design and use across several screens, then I want to be able to change design of the object with the information propagated to all screens containing it – so that I would not need to manually make the modification across multiple screens, and I could change the designs with a greater speed following User Feedback after User Testing sessions. | M |
| 5 | Cooperation with external project member | The project – Trilogue Tables – takes place in between the EP, the Council and at a later date the Commission. Part of the co-design work has already been carried out with the Council with the UXPin software used at the Council (Software #2). | S |
| 6 | Various Fidelity Levels | As a System Designer, I want to be able to progressively use higher fidelity, as the system design advance. | S |
| 7 | Comments | As a Testing User I want to be able to write my comments directly in the test environment. | C |
| 8 | Iteration handling | As a System Designer I want to bump +1 iteration, with the archiving occurring behind the scene to all the screens and the object that are inhered across multiple screens, so that I could work with greater ease of mind and have access to all the previous versions – all without manually managing files. | C |

\* Moscow Priority:

* **M** - Must: Describes a need/feature that must be satisfied in the software to be considered a success
* **S** - Should: Represents a high-priority item that should be included in the software if it is possible. This is often a critical need/feature but one which can be satisfied in other ways if strictly necessary
* **C** - Could: Describes a requirement which is considered desirable but not necessary. This will be included if time and resources permit.
* **W** - Won't: Represents a need/feature that stakeholders have agreed will not be implemented in a given release, but may be considered for the future.

## Assumptions and Constraints

### Assumptions

Assumptions are factors that are considered to be true, real or certain without proof or demonstration. Assumptions affect all aspects of planning and are part of the progressive elaboration of the initiative.

### Constraints

Constraints are those elements that affect the scheduling of an items or a restriction on the degree of freedom you have in providing a solution.

| **Constraints category** | **Description** |
| --- | --- |
| **Cost / Budget constraint** A cost constraint is any limitation or restraint placed on the project budget such as funds available over time |  |
| **Integration constraint** An EP system is depending of the proposed one and/or the proposed one depends on an existing EP system |  |
| **Compliance constraint** Is the requested activity required to comply with a compliance, regulatory or political requirement or constraint? | The software must be compliant with the EP standards and policies (cf [Environnement IT](http://www.standardsnet.ep.parl.union.eu/standards/cms/Accueil/infrastructure)) |
| **Continuity constraint** Is the requested activity required to ensure the continuity of existing business operations (e.g. maintenance or support of key business applications or infrastructure)? |  |
| **Short Term constraint** Is there a short-term deadline mandated for the execution/completion of the requested activity? |  |
| **Other constraint** An applicable restriction or limitation, either internal or external to the project, that will affect the performance of the project or a process |  |

# LSA - Software Proposals

List of potentially suitable software proposed by the LSA.

## Software #1

Please duplicate this section as many as suggested software.

In this chapter, you can suggest a software that meets some or all of your needs. When filling in this chapter, check the 'Liste des produits autorisés' (authorised product list) on the I[**Standards.net**](http://www.standardsnet.ep.parl.union.eu/standards/cms/Accueil/preconisations/P_Methodologie) site to see whether a software is already available[[1]](#footnote-1).

Software proposed by the requester:

| **Description** | |
| --- | --- |
| Product name  Version if required | Axure |
| URL/Website | http://www.axure.com/ |
| Give reasons for choosing the product you are suggesting: | A Desktop Installed – complex UI and interactivity of varying fidelities. |

If required, please provide the following information.

| **Complementary information  (**Service, Licence, Maturity, Architecture, etc.) | |
| --- | --- |
| Costs estimation | Licence(s): 895$ or 49$ per month.  Assistance (Consultancy, technical workshops, etc.): ?  Maintenance (30% Price): ?  Other (installation, upgrade, technical problems, etc.): Software testing, whitelisting, etc… |
|  |  |

**Comments**

If required, please provide the following information.

## Software #2

Please duplicate this section as many as suggested software.

In this chapter, you can suggest a software that meets some or all of your needs. When filling in this chapter, check the 'Liste des produits autorisés' (authorised product list) on the [**Standards.net**](http://www.standardsnet.ep.parl.union.eu/standards/cms/Accueil/preconisations/P_Methodologie) site to see whether a software is already available[[2]](#footnote-2).

Software proposed by the requester:

| **Description** | |
| --- | --- |
| Product name  Version if required | UXPin |
| URL/Website | https://www.uxpin.com/ |
| Give reasons for choosing the product you are suggesting: | Server hosted software for complex UI and interactivity of varying fidelities. |

If required, please provide the following information.

| **Complementary information  (**Service, Licence, Maturity, Architecture, etc.) | |
| --- | --- |
| Costs estimation | Licence(s): PRO licence 25euro per month per designer  Assistance (Consultancy, technical workshops, etc.): None  Maintenance (30% Price): None  Other (installation, upgrade, technical problems, etc.): None |
|  | The only cost associated with the solution is a monthly licence cost per Designer (unlimited test and presentation session can take place per every Designer licence). |

**Comments**

EU based company, software fulfilling all the needs for the project. Additionally the solution is already used at the other party collaborating in the interinstitutional project. The Council has already created assets that could be reused and shared with the EP with considerable savings on time and cost for the EP.

# CRM – Software Assessment

## CRM Cross-check

The request qualification could be modified from "Restricted to a user group" to "Corporate software" if other DGs have the same needs (internal CRM meeting).

***Request submitted to the other CRMs***

## EP software list check

Has an equivalent or similar software been authorised by the EP?

No

If Yes,

* Name the authorised product from the [authorised product list](http://www.standardsnet.ep.parl.union.eu/standards/cms/Accueil/preconisations/ingenierie-logicielle/liste_produits):
* Give the Software status:

If "Authorised - Unlimited usage", then justify why this product does not fit the needs:

If "Authorised - Restricted use", then justify why this product should be authorised for other users:

## Qualification

* **Software**:  Business

Technical

* **Usage of the product**:  Corporate software

Restricted to a user group

If checked, please precise for whom:

(Developers, etc.)

* **Support**:  By the DG ITEC/DIT

Included in the contract

* **Hosting**:  Local (on a desktop)

Departmental (administrated by the LSA)

In the EP Data centre (by DG ITEC/DIT/OPERATIONS)

Please note that for every requested product, they should be compliant with the [EP IT environment](http://www.standardsnet.ep.parl.union.eu/standards/cms/Accueil/infrastructure).

# Annex - Document control

## Validation

|  |  |  |
| --- | --- | --- |
| **Step** | **Date** | **Comment** |
| **Business Validation** |  |  |
| **LSA Validation** |  |  |
| **CRM Validation** |  |  |
| **STANDARDS Review (if required)** |  |  |

## Circulation

The content of the circulation table is not fixed and according to the request, the roles could change and user could be added.

| **DG or Company** | **Role** | **Name/Initials** | | **JIRA Role\*** | **ERICA[[3]](#footnote-3) code** |
| --- | --- | --- | --- | --- | --- |
|  | Sponsor |  |  |  | **A** |
|  | Requester |  |  | To be informed | R |
|  | Key user |  |  |  |  |
|  | LSA |  |  | To be informed | C |
|  | CRM |  |  | Reporter | C |
|  | STANDARDS |  |  |  | E/I |
|  | OPERATIONS |  |  |  | I |
|  | ... |  |  |  |  |

## \*in the SSPTRACK JIRA request

## Change history

| **Version number[[4]](#footnote-4)** | **Status[[5]](#footnote-5)** | **Date** | **Initials** | **Summary of changes** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Applicable documents

Note: Every listed document must be included as icon object

Applicable documents are standards, specimen plans which need to be applied and whose application is verifiable.

| **N°** | **Document name** | **Description** | **Document** |
| --- | --- | --- | --- |
| [1] | [Environnement IT Guide](http://www.standardsnet.ep.parl.union.eu/standards/cms/Accueil/infrastructure) | Presentation of the European Parliament's IT environment | Vos indispensables |
| [2] |  |  |  |

## Reference documents

Note: Every listed document must be included as icon object

Reference documents are a help or a support but are not directly applicable (e.g. software reference manuals, sample files from outside firms, meeting papers).

| **N°** | **Document name** | **Description** | **Document** |
| --- | --- | --- | --- |
| [1] |  |  |  |
| [2] |  |  |  |

## Glossary

| **Term** | **Description** |
| --- | --- |
| Business software | Business software is a software for the management of a specific business area (finance, employees, etc.). |
| CRM | In the DG ITEC, their role is to facilitate interactions between DG ITEC and its customers (Source: Presentation of CRM Services (CRMnet)) |
| Hosting | Various hosting are available:   * **Local hosting**: The software is hosted directly in the EP client desktop ("Fat client" type) * **Departmental hosting**: The software is hosted in the LSA infrastructure "datacenter" (client/server type). * **Datacenter hosting**: The software is hosted in the EP datacenter (client/server type). |
| STANDARDS | In the DG ITEC, their role is:   * Responsible for the strategy of platforms, i.e. maintaining an overall status and future plan of the architecture for development standards and tools * Developing and maintaining standard configuration   Source: <http://www.standardsnet.ep.parl.union.eu/standards/cms/Accueil/unite-standards> |
| OPERATION | In the DG ITEC, their role is to define and provide the IT infrastructure hosting the main applications of the EP (Source: EICInet) |
| LSA | In the GP and DG, their role is to Responsible for the ICT-functions (Information and Communication Technologies) of the DG/Political Group in question.(Source: MUST Welcome Presentation) |
| Products list (Standards.net) | List of the product by status ("standards with illimited use", "standard with restricted use", "non standards", etc.) |
| Requester | The requester is responsible to identify the need, test recommended software and decide about acquisition. |
| SSP | [Software Selection Process](http://www.standardsnet.ep.parl.union.eu/standards/cms/Accueil/preconisations/P_Methodologie/GEvalProd) |
| SSPTRACK JIRA | Request tracking tool dedicated to every software selection. |
| Support | Different supports are available:   * **Supported by the DG ITEC/DIT**: If the software is supported by the DIT, the following activities will be covered: helpdesk (relationship with the supplier), training, maintenance and release of the software. * **Not supported by the DG ITEC/DIT**: If the software is not supported by the DIT, the contract with the supplier must encompass support services (training, hotline, consultancy, etc.) |
| Technical software | Technical software is a software used for the development of applications or a component used by the applications running on the EP IT infrastructure and includes also all other software that is not a business software. |
| Usage of the product | Different software usage are available:   * **General use** product is a corporate software. Therefore, DIT supports the software * **Restricted use** products are for user group(s) only (not corporate) |

## Usage conventions

Where a chapter or section is not considered to be applicable, put "**NOT APPLICABLE**". Feel free to insert new sections (chapter, paragraph) as needed.

**Click on the Show/Hide  button in the toolbar to display/hide guidance.**

Guidance displayed.

1. Site Standards.net: [authorised product list](http://www.standardsnet.ep.parl.union.eu/standards/cms/preconisations/preco_outils_logiciels/liste_produits) / [Software Evaluation](http://www.standardsnet.ep.parl.union.eu/standards/cms/preconisations/P_Methodologie/GEvalProd) [↑](#footnote-ref-1)
2. Site Standards.net: [authorised product list](http://www.standardsnet.ep.parl.union.eu/standards/cms/preconisations/preco_outils_logiciels/liste_produits) / [Software Evaluation](http://www.standardsnet.ep.parl.union.eu/standards/cms/preconisations/P_Methodologie/GEvalProd) [↑](#footnote-ref-2)
3. **E**: Examen, **R**: Responsible, **I**: Informed, **C**: Contribution, **A**: Approval [↑](#footnote-ref-3)
4. Naming convention: Procedure 'Program & Project naming convention' ([**Standards.net**](http://www.standardsnet.ep.parl.union.eu/standards/cms/Accueil/preconisations/P_Methodologie)) [↑](#footnote-ref-4)
5. Status: Draft, Final, Approved [↑](#footnote-ref-5)