H2020-INSO-2014
INSO-1-2014 ICT-Enabled open government
YDS [645886] "Your Data Stories"

D6.1 Evaluation Methodology

Project Reference No	645886 — YDS — H2020-INSO-2014-2015/H2020-INSO-2014
Deliverable	D6.1 Evaluation Methodology
Workpackage	WP6: Pilots Deployment and Evaluation
Nature	R = Document, report
Dissemination Level	PU = Public
Date	01/09/2015
Status	Final v1.0
Editor(s)	Eric Karstens (EJC)
Contributor(s)	Mirko Lorenz (DW)
Reviewer(s)	Despoina Mitropoulou (GFOSS), Tilman Wagner (DW), Anna Triantafillou (ATC)
Document description	This document sets out the methodology and background for the evaluation activities related to *Your Data Stories* (YDS) during the three-year project period. Evaluation encompasses input from three continuous pilots, as well as from external test participants and experts. It is a staged process that includes operational feedback into the development process, but also assessments of usability, functionality, dependability, and acceptance of three definite prototypes, complete with a set of Key Performance Indicators (KPIs).

Document Revision History

Version	Date	Modifications Introduced	
		Modification Reason	Modified by
V0.1	28/06/2015	TOC	EJC
V0.2	30/07/2015	First draft	EJC
V0.3	07/08/2015	Consolidation of DW review	EJC
V0.4	31/08/2015	Consolidation of GFOSS review	EJC
V1.0	01/09/2015	Final Review & Submission	ATC

Executive Summary

This document describes the methodological approach to the evaluation of *Your Data Stories*, based on the best practice of the software sector, and taking the specifics of the project design into account.

The established principles of user-focused software evaluation and validation suggest that it is most effective and efficient to settle for a sufficiently high, yet limited number of tests and test participants. Moreover, user evaluation and validation should settle on a "good enough" and "fit for purpose" threshold in order to avoid feature creep and the emergence of ever new user requirements as the project develops. Only a focus on the original goals that is, however, flexible enough to accommodate essential adaptations of the original plan, will make sure that the project finishes successfully and on time.

More specifically, *Your Data Stories* takes a two-pronged approach to evaluation:

On the one hand, the project will establish three pilots, in which actual target users will apply YDS to real-world tasks. These pilots will create a permanent feedback loop with the developers even outside the formal evaluation cycle that will encompass three prototypes in years 2 and 3. In a manner of speaking, this is the "built-in" evaluation component of YDS that will help optimise the development process even ahead of, or in between, the planned prototypes.

On the other hand, the marked-off YDS prototypes will be tested in a dedicated fashion in order to establish the success (or not) of the project by way of clear qualitative and quantitative criteria (Key Performance Indicators, KPIs). This dedicated testing will, in turn, involve the pilot users and benefit from their detailed knowledge of the system, yet will also involve external users in order to stimulate unbiased, spontaneous feedback and better to explore usability and training aspects.

In addition to the above, the KPIs that are not directly user-focused will be assessed and demonstrated by experts from in- and outside the Consortium.

Table of Contents

List of Figures

List of Tables

List of Terms and Abbreviations

Abbreviation	Definition
YDS	Your Data Stories
D	Deliverable
WP	Work Package
KPI	Key Performance Indicator
FG	Focus Group
M	Month (of the project)
IEEE	Institute of Electrical and Electronics Engineers
GA	Grant Agreement for YDS
DoA	Description of Action

1 Introduction

1.1 Purpose and Scope

This report sets out the background, methodology and objectives for the user evaluation of the *Your Data Stories-Project (YDS)*. At the time of this report's drafting, the project is still in an early stage, i.e., several months away from the planned availability of the first integrated prototype, yet already in the middle of preparatory work. Accordingly, this document describes the evaluation methodology *ex ante*, based on the plans and intentions of YDS.

This is important in order to guide and plan both the development and the evaluation processes, which are to a great extent interdependent and will create a feedback loop: Users test and assess the project's (interim) achievements; their hands-on observations and critique provide further guidance as well as corrective suggestions to the developers. Users then evaluate the next, improved iteration, and so on. The three YDS prototypes that are foreseen in M12, M24, and M33, respectively, will also undergo a more formal evaluation that will check off both qualitative and quantitative criteria.

Seeing that any interactive software-based solution requires a certain amount of user training and/or instructions, user evaluations are also intended to yield insight into the most suitable approach to such training and possible tutorials. We will observe how users interact with the system and respond accordingly in creating suitable documentation and training materials such as online tutorials.

1.2 Approach to evaluation and relation to other work packages and deliverables

User evaluation is part of WP6, Pilots Deployment and Evaluation. This WP is described as follows:

The relevant tasks in this work package will ensure that YDS meets the demands of the intended user groups and that the tools and work processes provided are comprehensively picked up by the users from inception. They will also make sure that any relevant user comments and observations will feed back into the development process at a constant rate.[1]

This description indicates the specific concept of evaluation in this project: We will not merely survey users about the three marked-off prototypes to be delivered over the duration of YDS, but run permanent pilots similar to the "Living Lab"[2] approach in order to create and manage an ongoing mutual feedback process between targeted users and developers. This means that users will gain practical experience with YDS in parallel to the emergence of the full configuration of the system, rather than test it in bursts only upon completion of the respective prototypes. What we want to achieve with this approach is four-fold:

1) Continuous interaction between developers and users and thus increased ownership of the project on both ends;
2) Ongoing fine-tuning of the system in order to avoid issues even before they become manifest or difficult to correct;
3) Having a group of users available that is able to perform differential analyses of the system as it develops, i.e., recognise even incremental changes between versions; and

[1] Grant Agreement-645886-YDS_Final.pdf, p. 30
[2] https://en.wikipedia.org/wiki/Living_lab

4) Creating three seminal pilot user groups whose long-term experience with YDS may render them ideal to promote the system among their peers for future exploitation.

This evaluation methodology, and user evaluation itself, is therefore closely related to the other activities of YDS and involves all partners to some extent. However, most relevant for evaluation are the user partners, namely EJC (WP leader), GFOSS, MAREG, DW, D/PER, and NUIG. Results and insights gained will be shared and discussed with the technical partners, specifically ATC, TF, and NCSR-D.

In the run-up to this document, the user partners conducted focus groups in order to further clarify functional requirements and usage scenarios:

Focus Group No	Target Group
Pilot 1	
FG 1-1	NGOs and non-profit organizations and developers.
FG 1-2	Controlling and auditing bodies of the Greek Government.
FG 1-3	Businesses, Institutions and other organizations focusing on the constructions and energy business domains.
Pilot 2	
FG 2	Government, civil society, journalists and professionals.
Pilot 3	
FG 3	Data journalists and general-interest journalists.

Table 1: Focus Group Number and their relevant Target Groups[3]

The outcomes of these focus groups formed the basis for two relevant deliverables that were created under WP 2:

- D2.1 User Characteristics and Usage Scenarios v1.0; and
- D2.3 User Requirements v1.0

Both of these will be updated in the further course of the project (M12), and the updated versions will become the final starting point for the user evaluation efforts.

D2.1 details the user groups and their specific demands that will be addressed by the prototypes of YDS, namely the following (see Table 2)

[3] YDS Deliverable 2.3, p. 17, slightly modified for clarity.

User group	Description	Actions
Data journalists and general-interest journalists	Search for new stories and additional sources through simple and advanced options. Preview statistics and graphs. Download tabular data (raw and edited).	**Search** **Preview** **Download** Connect
Civil society	Actions related to transparency and accountability. First, issues relevant to the effectiveness and corruption are identified and then information is edited (comment, evaluate, share, edit content) in order to drive action in mass scale.	Search Preview Download **Edit** Connect
Auditors	Evaluate effectiveness & corruption in public bodies. Search is performed mainly through customized reports. In later stages, could be connected for customized reporting and data querying.	**Search** Preview Download Edit Connect
Web Developers	Web application development. Static and live data queries on YDS data could be the basis for building software or/and web apps.	Download **Connect**
Public institutions	Public institutions are interested in evaluating current and potential suppliers through their performance in public works. Evaluation could be both typical (e.g. law compliance) and related to operational efficiency.	**Search** Preview Download Connect
Suppliers	Search for business opportunities in public procurement. In later stages, could be connected for customized reporting and data querying.	**Search** Preview Download Connect

Table 2: Characteristics of the main user groups[4]

The deliverable then goes on to describe usage scenarios for each of the three pilots. These scenarios are intended to be used in the validation of the system, but may still be modified to a certain extent as the project moves forward:

Pilot 1 (responsible partners: NCSR-D, GFOSS, MAREG, and TF)

- Follow public money in Greece all the way
- Training and auditing (in order to improve scrutiny of public finances)
- The data citizen (public open data analysis from the perspective of citizens' interests)

Pilot 2 (responsible partners: EJC, DW, NCSR-D)

- Tracking development aid in the Netherlands

Pilot 3 (responsible partners: NUIG, D/PER, NCSR-D)

- Cross-Europe financial comparability

[4] YDS Deliverable 2.1, p.18, slightly modified for clarity.

The operation of the pilots will be elaborated in D6.2 Pilot Planning (M9). Each pilot will also report in a narrative fashion on their overall experience with YDS to both Pilot Monitoring and Evaluation Reports (M24 and M36, respectively).

D2.3, in turn, aggregates the user requirements based on the suggestions generated by the focus groups. At this point, it is still to be defined which specific user suggestions will be part of the eventual functionality of the prototypes.

Overall, the evaluation of YDS will put the abovementioned scenarios into practice. The evaluation of the pilots will hence help to capture suggestions for the system's development and, in collaboration with external evaluators, eventually help to verify whether the system's functionality, usability, and acceptance are adequate.

1.3 Methodology and Structure of the Deliverable

This deliverable is based on best practice developed during previous projects – namely CASAM (FP7 Grant Agreement 217061, 2008-2011) and SYNC3 (FP7 Grant Agreement 231854, 2009-2012) –, on a brief review of relevant literature and the approaches to evaluation of similar projects, and on the customisation of the methodological approach to the specific demands of YDS.

This document first reviews the underlying principles of validation and evaluation in order to set the scene and to establish the prevailing view on the number of evaluators and tests. It will then elaborate on the specifics of YDS with its mixture of continuous pilots and dedicated prototype tests, establish the Key Performance Indicators for the project. Finally it will explain the practical approach aimed for to evaluate the YDS' outcomes.

2 Principles of validation

This section provides an overview of why and how testing in software development is necessary, as well as of which procedures create value, ensure quality and enhance visibility of progress towards the final product.

Questions to be answered are:

- What is the benefit of testing and validation in software development?
- Which best practices can be identified?
- How can the findings be applied to test and validate the YDS system?

The research reviewed in this section shows that a number of insights can help significantly if applied to this particular project. The following pages provide an overview of the factors relevant to the software development process in general and particularly for testing and validation.

At the end of each sub-section the relevant insights for YDS are highlighted, so as to provide an overview of the practices that will be applied during the actual evaluation process, which is described in Sections 3 and 4.

2.1 Definition of user validation

2.1.1 Validation: Assuring that a software system meets the user's needs

The statement "Assuring that a software system meets the user's needs" is a short and easy to remember definition of validation. A clear understanding of the term is important, because wrong assumptions of what it means may lead to confusion regarding the specific scope of testing and validation.

For example, a definition from a technical standpoint that describes validation as "the process of evaluating software at the end of the software development process to ensure compliance to software requirements", is too narrow. Starting to validate once the project nears completion would be too late to uncover and still correct flaws, usability issues and integration issues. Therefore, the following paragraphs will discuss the meaning of validation from different angles.

Multiple sources such as dictionaries and manuals put it this way: "In general, validation is the process of checking if something satisfies a certain criterion. Examples would be:

- checking if a statement is true (validity);
- if an appliance works as intended;
- if a computer system is secure, or
- if computer data are compliant with an open standard.

Validation implies one is able to testify that a solution or process is correct or compliant with set standards or rules."

2.1.2 The definitions of verification and validation for the purpose of YDS

The definition that we use to describe and understand the scope, the desired outcome of the testing and validation process is the one below, differentiating between the objectives from two points of view:

- **Validation: "Are we building the right product?"**
  i.e., does the software meet the end users' actual needs?
- **Verification: "Are we building the product right?"**
  i.e., are there defects or bugs in the code?[5]

As for the difference between "validation" and "verification", it is important to note that both are related, yet very different concepts, and require specific procedures to be applied properly. ***Verification*** essentially means to ensure that the software has no serious defects or flaws, such as software bugs.

It should be noted that tracking and resolving defects is not in the scope of the particular task discussed in this document. As mentioned above, the user validation tests will be used to look at the modules from a user perspective. Ensuring that the modules work from a technical view will be a task of their respective technical developers. However, should any bugs be uncovered during the validation process, they will be immediately communicated to all stakeholders affected.

2.1.3 Relevant validation standard

As a formalized process, standards defined by IEEE describe processes to ensure quality standards. Proper validation is a tool in the development process in order to reach both a level of confidence that the product fulfils the user needs, as well as meets so-called "minimal" criteria of quality standards.

The most relevant standard is IEEE 1012-2012 (Standard for System and Software Verification and Validation), which provides an outline covering relevant aspects of validation. This standard is part of a whole system of standards applied to software development from different perspectives and at different stages of development.

The relevant IEEE definition of validation is the
"Confirmation by examination and provisions of objective evidence that the particular requirements for a specific intended use are fulfilled."

More specifically, the validation process
"Provides supporting evidence that software satisfies system requirements allocated to software and solves the right problem. "

In order to achieve this, the standard foresees the development of a customized methodology and evaluation plan as well as the definition of a baseline (*status quo ante*) against which the eventual system as well as its intermediate versions can be compared.

2.2 The role and relevance of validation

Software development is an area of knowledge that keeps gaining importance. The reason for this is the dependency of businesses, consumers and public institutions on software-based services which work without major flaws. Another driver is the need to ensure the quality of ever more complex software systems. As a result, any knowledge that supports the development of innovative, productive and easy-to-use software is becoming an important differentiator in competitive economic environments as well as public services.

[5] https://en.wikipedia.org/wiki/Software_verification_and_validation

Against this background it is easier to understand why the software creation process has been the target of many studies and research projects that tried to gather knowledge of how to achieve desired project outcomes. Statistically, many software projects run into trouble along their timeline. Even today, the number of failed or challenged software projects runs up to 63%[6]. There are numerous reasons for this, the main issue being complexity.

The most complete set of data tracking the success of software projects is supplied by the Standish Group, a US research firm that has conducted extensive research in order to determine success rates of software projects of various sizes over many years. Since 1994, the Group publishes an annual study called the CHAOS Report, which is based on statistical data from tens of thousands of software projects of various scales. From this extensive set of data, the Standish Group developed general recommendations to ensure software development success. The key question is how many projects finished as a success or failure, or remain precarious while already in use.

Category	Description
Successful	The project is completed on time and on budget, with all features and functions originally specified.
Challenged	The project is completed and operational, but over budget, late, and/or with fewer features and functions than initially specified.
Failed	The project is cancelled before completion, or never implemented.

Table 3: Success or failure of software projects[7]

The good news is that in recent years, success rates have an upward tendency, while cost and schedule overruns are declining. The CHAOS research timeline (see Figure 1) provides evidence of overall, if precarious, improvement in IT project management.

	2002	2004	2006	2008	2010
Successful	34%	29%	35%	32%	37%
Challenged	51%	53%	46%	44%	42%
Failed	15%	18%	19%	24%	21%

Figure 1: Multi-annual success statistics[8]

2.2.1 Success factors in software development

The CHAOS report series identifies ten success factors, called the “CHAOS 10”. This list provides an overview of which factors have a high positive or negative impact on software projects. Although no project requires all 10 factors to be successful, the more factors present in the project strategy, the

[6] The Standish Group: CHAOS Manifesto 2012, p. 3
[7] Ibid.
[8] Ibid.

higher the confidence level. Used proactively, these factors can be viewed as early indicators to avoid failure and as a means to define priorities early on.

User involvement, which is the key task of evaluation and validation, is ranking high on the list. It is the second most important factor to ensure successful software development and has been the most important factor in past studies.

Recipe for Success: CHAOS 10 Each factor was weighted according to its influence on project success. The more points, the higher the impact.		Success factor
1	Executive management support	19
2	User involvement	18
3	Clear business objectives	15
4	Healthy, constructive development environment	12
5	Optimization of the development process	11
6	Agility of the process (feedback loop with users)	9
7	Project management expertise	6
8	Skilled human resources	5
9	Execution and control of the development process	4
10	Tools and infrastructure	1

Table 4: Recipe for Success: CHAOS 10[9]

Relevant finding for YDS:

> ***The CHAOS reports confirm the paramount importance of user involvement and a feedback loop with the developers in the software development process. Well-implemented rounds of user testing and evaluation will therefore not only serve to assist the engineers in optimizing software functionality and usability, but also enhance the no less crucial end-user acceptance of the eventual product.***

2.3 Planning the validation process

There are many different ways in which validation tests can be planned and applied. A short overview helps to identify the testing process best applicable to YDS.

2.3.1 Basic methods of evaluation

Designing a system test forces a development team to deeply understand the requirements. The better these requirements are visible, the earlier incompleteness, ambiguity, and inconsistency can be

[9] Ibid., p. 4; modified to improve clarity

identified. Correcting such problems early will speed up development and reduce the number of late requirements changes.

There are three basic methods of evaluation:

1) **Ad-hoc testing**
- "Just see if you can break it";
- Make up test cases "on the fly";
- Human interpretation of requirements.

This method is of particular relevance for the external evaluation rounds. Users who are not accustomed to working with YDS are asked essentially to "play around" with the system in order to discover usability flaws as well as functionality and acceptance issues.

2) **Systematic testing**
- Driven by explicit quality assurance goals;
- Tests designed for comprehensive coverage;
- Tests specify expected output as a benchmark.

This method is of particular relevance for the pilots, which will deal with specific, pre-defined use scenarios, and thus focus user testing on a limited, controlled amount of tasks and data sets with which the users are familiar. In such a way, even nuances can be detected.

3) **Automated testing**
- Driven by explicit quality assurance goals;
- Test-suite designed for comprehensive coverage;
- Scripts need no human judgment.

This method is not relevant for YDS, because the entire system usability and benefits hinge on user interaction and user-driven intention.

Relevant finding for YDS:

> ***The innovative character of YDS with its focus on the integration of previously separate work processes and a variety of data sources requires a well-balanced mix of basic evaluation methods in order to cover the full range of evaluation needs.***

2.3.2 Validation and verification

There is a strong dependency between validation and verification. Verification must be performed regularly to identify and eliminate flaws and defects of the software. This task is usually very complicated when a large project is partitioned into different modules that need to be integrated at some point during the process.

Validation can identify whether a user can execute a task with or without training. It can answer the question whether the graphical user interface (GUI) is intuitive or needs detailed and sophisticated help documentation. For example, a test can verify whether a user is able to perform a search and retrieve

the data that he/she is interested in, and use the various options that the system provides for exploring data: Relations between data sets, getting an overview of relevant and valid information related to a topic, exploring the data based on time, location, causal relations, etc., visualize data and findings based on data, add comments and opinions to them and involve the "crowd" by way of social media, and so on.

Validation therefore is not only a tool to create user-friendly interfaces. Complex tasks may well require complex software with many features that are not necessarily intuitive and self-explanatory. Validation is also a very effective way to learn which training and documentation is needed in order to gain acceptance.

Verification, on the other hand, is needed to ensure that the software runs without major defects, e.g., that large files can be processed without stalling the system, or that stored data is secure and can be found reliably when a search is initiated. Verification can be executed as a static or a dynamic process. Static verification usually refers to a software inspection or a code analysis. Dynamic verification is executed with test data to check how the system is working under load.

Relevant finding for YDS:

> ***The dependency between validation and verification is an issue that must be covered in the test plan. Validation and verification are two sides of the same coin. Verification must be completed to a high degree before each round of validation, while validation will provide feedback for modifications, which will, in turn, require verification before user testing.***

2.3.3 Level of confidence

The "level of confidence" describes a status when many foreseeable factors are under control. This relates to different aspects such as functionality, integration, time, budget, etc. The term is also helpful to keep development confined to the key features and avoid the uncertainty that is caused by too many changes.

Finishing a project when it is "good enough"
Verification and validation establish confidence that the software is fit for its purpose. This term applies strictly to a particular version or release of software in the development lifecycle. Therefore, this process will normally not result in software free of defects or including every conceivable feature. More to the point, the goal is to be able to finish a project and get to a product that is "good enough". This level of quality is usually the goal of standards such as IEEE or ISO, which describe the minimum level of quality that is needed.

Releasing a system when it is "fit for purpose"
The required level of confidence can vary from system to system. High-security software needs a higher level, while broadly used systems must ensure that, e.g., user data or stored content cannot be altered. Therefore, it is important to define a set of criteria that describes what to test during a validation and verification process. The specification usually referred to in addition to "good enough" is usually to test whether a system is "fit for purpose".

Both terms are important to ascertain a certain level of flexibility in the development process, and they open productive ways to turn ideas into features from release to release.

Relevant finding for YDS:

> ***The goal of creating a system that is "good enough" and "fit for purpose" is important for user evaluation as well. These concepts are actually a warning not to use testing to search for more features that could be included. This would create the risk that the project becomes open-ended. In order to avoid this, a "wish list" will be one of the tools to collect possible features for future releases of YDS without compromising the ongoing development.***

2.3.4 Test early and often

The CHAOS 10 identifies "user involvement" as a key success factor. It does not define how it is supposed to be initiated, planned, and managed. What can be done to really uncover user requirements while they shift or change, particularly in the field of open data?

The point is that the only way to get to a level of certainty is to involve users in different stages of development and to use a set of tools to ensure that the needs of these users are understood in the best possible way. This is an iterative process. Results from interviews, use cases and use models can result in surprises and new directions, which can shift the demands regarding development. The less momentous these new insights are, the easier it is to finish a project in time and on budget.

Validation and verification therefore must be applied in each stage of the development process. Simply testing an end product at a late stage will statistically often result in major re-work demands.

Relevant finding for YDS:

> ***Constant user involvement is the key. Testing early and often will create additional benefits. This is why YDS is using pilots in order to capture feedback at short notice during the entire development process in addition to the testing of the prototypes.***

2.3.5 Number of tests and evaluators

How many users must take part in tests in order to create a system that is "fit for purpose"? Usability expert Steve Krug[10] presents compelling data and findings that testing does not need to be overly complex in order to get results which are helpful for the development process and the programmers.

His recommendations, as well as suggestions from other authors[11] are helpful as they provide background information on how the test process can be set up and how many users must be tested to achieve trustable results. Additionally, simplifying testing as much as possible helps to reach the goal of testing "early and often", which – as discussed above – is important to ensure project success. Krug's

[10] Steve Krug: Don't Make Me Think. A Common Sense Approach to Web Usability. Second Edition. Berkeley, CA, 2006

[11] For a recent meta-study overview of similar research, see Claudia Zapata, José Antonio Pow-Sang: Sample size in a heuristic evaluation of usability. In: Software Engineering: Methods, Modeling, and Teaching. Volume 2, Lima 2012, pp. 37-46

book, however, is about usability, not validation. Although validation goes further and is more formal than usability testing there are still many aspects relevant to both disciplines.

A distinction to be made is that validation judges interface design based on formal requirements as well as from a functional perspective, yet not expressly from a marketing point of view, where, for instance, an optimised user interface may be highly important (e.g., in the case of e-commerce). Usability in a wider sense is focused on developing the best possible user experience for end users in order to enable them to execute their tasks efficiently, effectively, and in a comfortable fashion. If that is the case, it obviously also represents a marketing advantage. Hence, marketing-oriented usability research provides approaches and findings that are valuable for the usability of a product as a whole. Even though there is a difference in the main intention between user interface evaluation and "holistic" user-centric evaluation, there is no categorical difference.

The following excerpts[12] from *Don't make me think* provide relevant information on which aspects need to be considered:

The importance of recruiting representative users is overrated.
It's good to do your testing with people who are like the people who will use your site, but it's much more important to test early and often. My motto (…) is 'Recruit loosely, and grade on a curve.'

The point of testing is not to prove or disprove something. It's to inform your judgment.
People like to think, for instance, that they can use testing to prove whether navigation system "a" is better than navigation system "b", but you can't. No one has the resources to set up the kind of controlled experiment you'd need. What testing can do is provide you with invaluable input which, taken together with your experience, professional judgment, and common sense, will make it easier for you to choose wisely – and with greater confidence – between "a" and "b."

Testing is an iterative process.
Testing isn't something you can do once. You make something, test it, fix it and test it again.

Nothing beats a live audience reaction.
One reason why the Marx Brothers' movies are so wonderful is that before they started filming they would go on tour on the vaudeville circuit and perform scenes from the movie, doing five shows a day, improvising constantly and noting which lines got the best laughs. Even after they'd settled on a line, Groucho would insist on trying slight variations to see if it could be improved.

[12] Krug, p. 135

	TRADITIONAL TESTING	LOST-OUR-LEASE TESTING
NUMBER OF USERS PER TEST	Usually eight or more to justify the set-up costs	Three or four
RECRUITING EFFORT	Select carefully to match target audience	Grab some people. Almost anybody who uses the Web will do.
WHERE TO TEST	A usability lab, with an observation room and a one-way mirror	Any office or conference room
WHO DOES THE TESTING	An experienced usability professional	Any reasonably patient human being
ADVANCE PLANNING	Tests have to be scheduled weeks in advance to reserve a usability lab and allow time for recruiting	Tests can be done almost any time, with little advance scheduling
PREPARATION	Draft, discuss, and revise a test protocol	Decide what you're going to show
WHAT/WHEN DO YOU TEST?	Unless you have a huge budget, put all your eggs in one basket and test once when the site is nearly complete	Run small tests continually throughout the development process
COST	$5,000 to $15,000 (or more)	$300 (a $50 to $100 stipend for each user) or less
WHAT HAPPENS AFTERWARDS	A 20-page written report appears a week later, then the development team meets to decide what changes to make	The development team (and interested stakeholders) debrief over lunch the same day

Figure 2: Traditional vs. simple testing[13]

2.3.6 Number of users

Steve Krug suggests the ideal number to be three or four users for each round of testing. The first three users are very likely to encounter the most significant problems. He argues that 3-4 users in two rounds of testing will discover more of a software or site's usability problems than 8 users in a single round, and that there are diminishing returns for testing additional users.

[13] Krug, p. 137, slightly modified for clarity

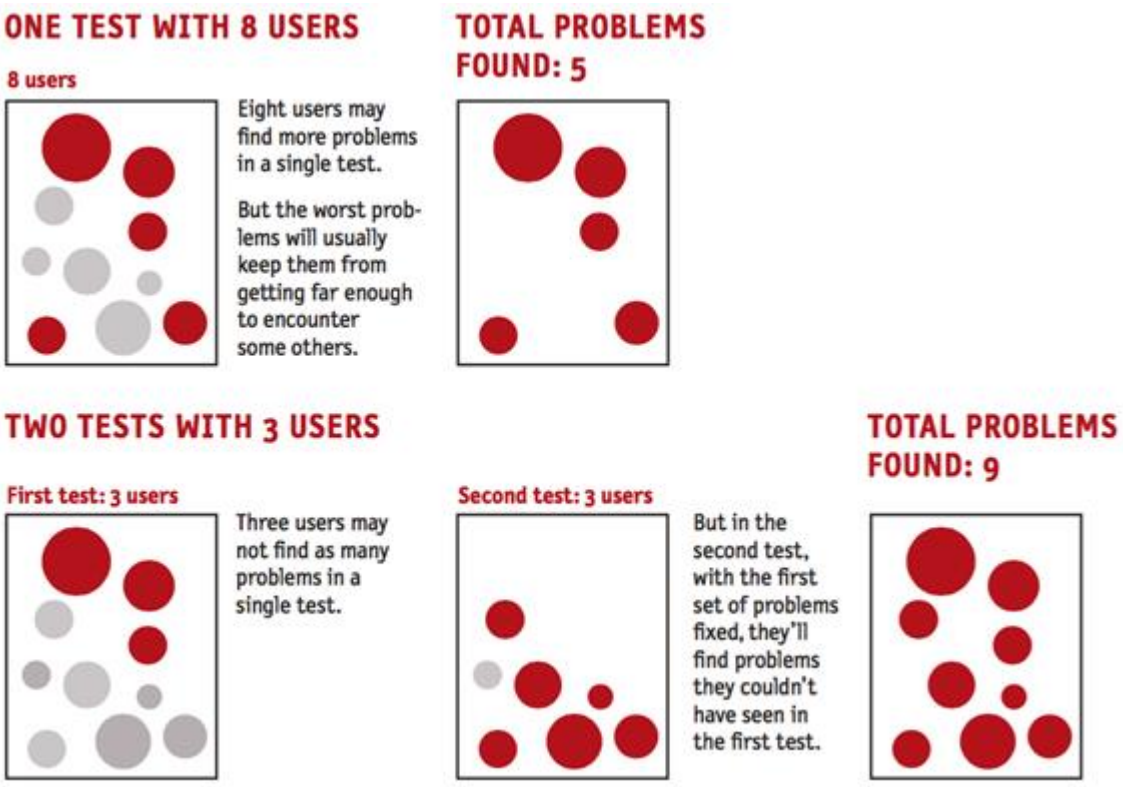

Figure 3: Three users will discover 85% of an application's usability problems[14]

Many discussions have taken place about the sample sizes for usability evaluation since the studies of Steve Krug and others. One of the most popular rules in the usability evaluation field at the present moment is the "4±1"or "magic number five" rule.[15] Nielsen and Molich[16] studied the issue of sample size for usability testing in the case of the heuristic evaluation method, in which a small number of evaluators inspect the user interface design of a system in order to detect usability problems in comparison with known usability principles (heuristics).

By means of empirical studies and mathematical methods the authors demonstrated that 2/3 of usability problems are detected by five evaluators, and that tests with additional evaluators are not likely to expose new problems. Virzi[17] indicated similar findings in the case of determining the optimal sample size for usability testing using the "think aloud" method (see below). He demonstrates that four or five users detect 80% of usability problems.

These findings have been challenged by several subsequent studies[18]. A meta-analysis of sample size issues in usability evaluation conducted in 102 usability evaluation experiments extracted from online academic databases, including the ACM Digital Library, IEEE Xplore, and ScienceDirect, and offline sources since 1990, indicated that in order to obtain reliable/optimal results using the "think aloud"

[14] Krug, p. 139
[15] Wonil Hwang, Gavriel Salvendy: Number of People Required for Usability Evaluation: the 10±2 Rule. *Communications of the ACM* 53.5 (May 2010), p. 131
[16] Nielsen and Molich, quoted ibid.
[17] Virzi, quoted ibid.
[18] Law, L. -C. and Hvannberg, E.T. "Analysis of combinatorial user effect in international usability tests." *CHI Conference on Human Factors in Computing Systems*, ACM (2004): 9–16; Slavkovic, A. and Cross, K. "Novice HEs of a complex interface." *CHI '99 extended abstracts on Human Factors in Computing Systems*, ACM (1999): 304–305

method, the size of the user sample must be nine in order to detect 80% of the usability problems. This is also corroborated by a more recent meta-study[19], which asserts the number of 10±2 users as the gold standard.

Relevant finding for YDS:

> ***Wherever possible and appropriate we will apply the recommendations provided above to simplify the test process. The main reason is that this will result in more testing within a shorter time period and test results that can be used quickly by the developers. A complicated test process with high user numbers and a formal test report would not fit the dynamic nature of this particular, current affairs-driven project, and would lack in efficiency.***

[19] Zapata, Pow-Sang, ibid., p. 45

3 Specific evaluation framework conditions for YDS

Your Data Stories takes a specific approach to development and must be evaluated at various levels. This section explains how and why.

3.1 YDS objectives to be evaluated

As per the DoA[20], the goal of YDS is defined as follows:

> ***"YourDataStories" (YDS) is a highly customisable online platform for data exploitation focused in the financial flows that are critical for transparency, collaboration and participation, all pressing social challenges ranked highly in the European agenda. Users are facilitated by powerful and established tools, not only to discover relevant information but also to remix it with diverse and dynamic data sources: YDS acts like an interactive canvas to enable data citizens to (re)write their own data history.[21]***

More specifically, YDS' overall objectives are:

Objective as per the Grant Agreement	Description
1) Design a unifying conceptual model	Render open government data from a variety of sources comparable and interoperable with each other through a single user interface
2) Redistribute data under a unifying conceptual model	Become a one-stop-shop for open government data so that users do not need to visit (or search on) a variety of portals
3) Enhance open government data	Cross-check multiple data sources in order to detect errors, mistakes, and missing information, and connect previously unconnected datasets to provide added value
4) Add a social dimension to open government data	Use the social and semantic web to enhance, enrich, and identify meaning of, and relations between, data (social media, linked data)
5) Increase governmental transparency through increased open data visibility	Make open government data available for easy public scrutiny and public comment
6) Exploit the social dimension to augment open government data with user-generated content	Use social media to comment on and curate open government data
7) Create an open ecosystem for marketing data, services, and applications	Enable the creation of third-party applications based on YDS

Table 5: YDS objectives as per the Grant Agreement[22]

Not all of the above are directly relevant for user evaluation, but rather represent technological and/or exploitation objectives, which will, however, be covered by the evaluation reports (D6.3 and 6.4).

[20] GA, p. 7-9
[21] Ibid., p. 5
[22] Ibid., p. 7-9

3.2 Feedback loop with pilots

In contrast to many other software projects, YDS integrates professional users from the start and in an ongoing fashion. This means that, obviously, the target user groups already have played a crucial role in the definition process of functional requirements for the eventual system as presented in Deliverables D2.1 and D2.3. However, users will remain involved for the majority of the project period through the three YDS pilots:

1. Follow public money;
2. Tracking development aid in the Netherlands; and
3. Cross-Europe financial comparability.

The primary goal of the pilots is to put any version of YDS to the practical test as they become available, in order to demonstrate that they are fit for purpose, while at the same time providing a constant stream of feedback on usability, performance, errors, and real-life viability to the developers. The pilots will also be used to evaluate the three distinct prototypes.

The secondary goal is to support YDS' dissemination activities. This is because the best advertising for the project is tangible impact in the real world, such as journalistic coverage, better-informed political decision-making, or citizen awareness of the pilot topics.

According to the findings of the studies described above, it is advised for each pilot is set to have on average around 10 active users. The user groups will be composed in such a way that they best reflect the focus of the respective pilot, e.g.:

Follow public money in Greece	Tracking Dutch development aid	Comparative analysis of public finances in Greece and Ireland
• 6 public officials • 2 citizens/NGO representatives • 2 journalists	• 6 journalists • 2 public officials • 2 citizens/NGO representatives	• 6 public officials • 2 journalists • 2 economists

Table 6: Composition of the pilot user groups

All users should be properly qualified, i.e., be experts or stakeholders in the respective fields. While their nationalities do not necessarily matter, it is still advisable to include at least 2 citizens or residents of the countries in question who are familiar not only with the local language, but also with the prevailing framework conditions. The use of the partners' staff members is fine as long as their numbers do not exceed one third of the entire group. Due to its thematic diversity, pilot 1 may require a larger group of persons to be involved.

The pilots will run for the better part of the entire lifetime of YDS, i.e., from autumn 2015 until the end of the project in January 2018. This is to ensure that the Consortium is able to

- record benchmarks of the status quo before YDS;
- collect user feedback on usability as well as functionality during the development process (i.e., while changes may still be considered and implemented);
- document the progress and innovation brought about by YDS as it emerges;

- prove the viability and fitness-for-purpose of the intermediate and final prototypes.

In synch with the development of the YDS project itself, the pilots will be composed of three phases:

- determination and documentation of benchmarks (year 1);
- phasing-in of YDS usage and providing feedback to developers (year 2);
- full-fledged practical use of YDS and final user evaluation and validation (year 3).

3.2.1 Phase 1: Baseline

Phase 1 (year 1) will define specific use cases based on the needs and interests of the users. At least half of the use cases should have long-term relevance. Pilot participants will investigate the effort such use cases would require without the help of YDS. Where possible, participants might already have implemented similar projects and use their experience as guidelines. This includes research into suitable data sources and their assessment, as well as the scrutiny of available tools to process and visualize the data. Social media and crowdsourcing options should be considered as well.

Outputs of phase 1: Benchmarks for the final user evaluation and validation; refined user requirements for YDS.

3.2.2 Phase 2: Early versions and prototypes

Phase 2 (year 2; first and intermediate prototype) will start (re-)implementing projects such as the ones defined in phase 1 with the assistance of any emerging YDS components and the early integrated system. At this early stage, the main focus will be on optimizing work processes with YDS in terms of usability and functionality. Most likely, this will necessitate a rather intensive exchange with the developers in order to resolve any issues as they become apparent.

Outputs of phase 2: Rapid feedback on usability and functionality of first year prototypes, essentially putting users in direct contact with YDS' developers.

3.2.3 Phase 3: Advanced versions and prototypes

Phase 3 (year 3; intermediate and final prototype) will focus on the full-fledged, mature implementation of the use cases with the aim of delivering actual outputs (journalistic coverage, studies, interactive graphics, etc.), which are better than the ones that were realistically feasible without YDS. Interaction with the developers will focus on ad-hoc trouble-shooting and cooperation on the integration of the various components of YDS. Users will be observed when using the platform and will be asked for express feedback (focus group style).

Outputs of phase 3: Final user evaluation and validation of all KPIs; projects set up to continue beyond the YDS project lifetime.

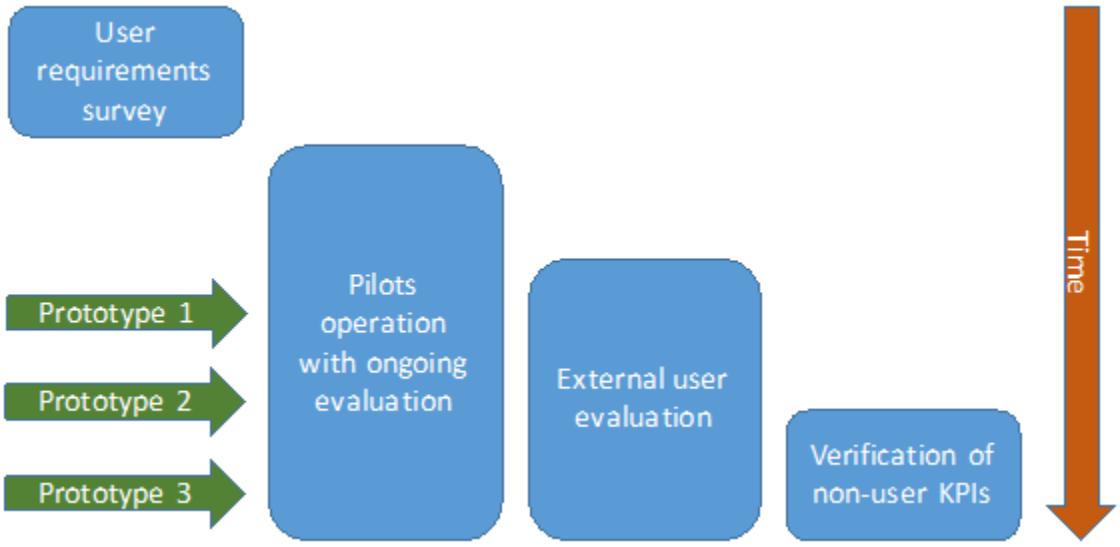

Figure 4: User involvement in the YDS development and evaluation process

3.3 External user evaluation

While we believe that the three pilots, which are run very close to the project and are meant to stimulate a permanent conversation between developers and users, will provide extremely useful and timely feedback, we also expect a certain amount of identification to occur between users and the project. This may bias their assessments or render them blind for potential flaws they may have gotten used to. Hence we will also evaluate each of the three pilots with external members of the relevant user groups. "External" here means persons without direct attachment to the project or any of the pilots; they may still in part be recruited from the user partners' staff.

Each prototype will be tested with between 5 and 9 external users per pilot. It is desirable to have the same set of users assess all three pilots in order to enable differential analysis; hence, all partners involved should try and keep attrition to a minimum.

3.4 Non-user related success indicators

Some of the success indicators of YDS do not directly relate to user interaction or user requirements, but measure technical achievements or the system's potential for exploitation. Accordingly, they require an assessment on a more abstract level, for instance in consultation with subject-matter experts. An example would be the integration capacity and state-of-the-art of ontologies used by YDS[23].

3.5 Key performance/success indicators

In line with the above, it is necessary to define three sets of indicators that optimally reflect the parallel approaches: For pilot users, external target group representatives, and for the non-user related aspects. All benchmarks must be achieved by the end of the project period. Lower values at the time of the first or interim prototype are to be expected and do not indicate failure, but rather reflect the development and improvement process. Similarly, it should not be assumed that KPIs develop in a linear fashion over the three years of YDS.

[23] GA, p. 5

3.5.1 Indicators for pilot users

Persons involved in the pilots have a special relationship with YDS. First, they will establish ex-ante baselines to compare the eventual system against. Second, they are involved in an ongoing fashion, and thus will provide continuous feedback to the developers even outside the formal evaluation schedule. And third, they will become experienced with the system at an early stage, which means that they will not be entirely unbiased after a while. This translates into the following indicators:

Indicator	Metric
Benchmarks/performance	Benchmarks exceeded by at least 20% (e.g., time saved)
Usability/ease of use	Rated good or excellent on a 5-point scale by at least 60% of respondents
Functionality (as per the User Requirements)	Rated good or excellent on a 5-point scale by at least 60% of respondents
Presence of new, innovative features	At least 80% of respondents acknowledge 5 or more innovative features (unprompted)
Dependability	No system crashes experienced while using the system
Overall satisfaction	Rated good or excellent on a 5-point scale by at least 60% of respondents

Table 7: Indicators for pilot users

3.5.2 Indicators for external users

External users are much less familiar with the system or its user scenarios. Accordingly, they provide a better gauge for the intuitive usability of the system because they are less attached to it – i.e., likely more critical. They also have no systematic way to establish a baseline, but will still compare YDS to other systems they have used. This translates into the following indicators:

Indicator	Metric
Usability/ease of use	Rated good, excellent, or fair on a 5-point scale by at least 60% of respondents
Functionality (as per the User Requirements)	Rated good, excellent, or fair on a 5-point scale by at least 60% of respondents
Presence of new, innovative features	At least 50% of respondents acknowledge 4 or more innovative features (unprompted)
Dependability	No system crashes experienced while using the system
Overall satisfaction	Rated good, excellent, or fair on a 5-point scale by at least 60% of respondents

Table 8: Indicators for external users

3.5.3 Indicators for non-user related success criteria

These are the overarching indicators defined in the DoA[24]. Measuring these cannot be achieved through a questionnaire or by observing users, but must be determined and demonstrated by experts familiar with the system and its exploitation. This translates into the following indicators:

Indicator	Metric
Improvement above state-of-the-art in ontologies, vocabularies, and thesauruses concerning open government data.	Ontology accommodates all contemporary open government data related to the thematic domain of YDS (yes/no)
Ability to query and retrieve multiple sets of open government data through a single end-point (user interface). Queries to be defined during the development process.	YDS is able to answer queries that cannot be answered by any single open government data source (yes/no)
Ability to query and retrieve enhanced aspects of the open government data contained in the YDS repository. Queries to be defined during the development process.	YDS is able to answer queries that cannot be answered correctly by non-enhanced open government data (yes/no)
YDS successfully connects open government data with the social web by way of links and connections between the social and semantic aspects of open government data.	Social media interaction and/or pilots have created at least 50 semantic-to-social and/or social-to-semantic links.
Usefulness of user-generated enhancement of open government data through YDS.	More than 50% of additions to open government data by way of the social web and/or the pilots are valid and useful.
Pilot applications enhancing transparency and the fight against corruption, built on top of YDS, are favourably evaluated.	*See sections 3.5.1 and 3.5.2 above.*
Pilot applications have performed their role in communication and marketing of YDS.	At least two governmental bodies or civil society organisations have taken to using any of the pilot applications on their own accord.

Table 9: Indicators for overall KPIs

[24] GA, p. 7-9

4 Practical user evaluation methodology

The YDS user evaluation methodology presented here is a systematic approach to ensure that the requirements of YDS are met in the final product and that the prototypes released will be properly validated.

Based on goals and best practices identified to be useful, this report is meant to become a tool for the next phases of this project. Key points are a good understanding of the requirements as well as a detailed workflow.

4.1 Scope of testing

The entire YDS user feedback activities are designed to enable a broad range of validation procedures, including tests of basic material such as layouts, wireframes, rough sketches of user interfaces or functional modules, or simulations, as well as the prototypes of the complete system. This flexibility is needed to respond to the needs of the developers. The perspective is that any interaction with users can produce new insights how to build and optimize the final system.

However, in terms of validation the tests naturally become more meaningful as the system completion and especially integration progresses, i.e., once the prototypes are on hand. Otherwise the risk of testing incomplete material might result in not detecting issues or user frustration and less willingness to continue supporting the project.

As discussed above, the validation aims to ensure that "the right product will be built" from a user perspective. In case that the modules do not perform the desired tasks in a test situation for technical reasons, the test result will be labelled as "failed", and a note will be sent to developers asking for verification of the code.

4.2 Prototypes and evaluation scenarios

What will be evaluated in particular? This section provides a description of the suggested test and validation scenarios for the three prototypes of YDS, as well as for interim versions:

- Ongoing practical hands-on testing for the three pilots;
- First integrated prototype (M12), which is planned already to implement a comprehensive scope of YDS' components;
- Second integrated prototype (M24), which will be based on further improved components;
- Final integrated prototype (M33), which will be even further improved so as to become the springboard for post-project development.

Obviously, the three main rounds of user evaluation need to take the different stages of completion into account. The first prototype will be checked primarily for basic functionality and general user understanding of the YDS practical purpose and usability. In the second and third rounds, testing must be done as comprehensively as possible regarding the functional requirements and in order to validate them one by one with the different targeted user groups.

This means that we will check all focus group user propositions and user requirements as set out in deliverable D2.3 (and later on, as updated in deliverable D2.4) against qualitative user feedback or other relevant indicators.

4.3 User evaluation methodology

In the course of the YDS user evaluation, we employ tried-and-tested techniques to elicit useful feedback from the test persons. To this end, and different from the focus group brainstorming sessions during the user requirements analysis, user evaluation is usually conducted in face-to-face sessions with a maximum of four persons present:

- The up to two test leaders, asking questions, giving instructions and assignments, debriefing, observing and taking minutes;
- The primary test user, performing the evaluation and, where applicable, subsequently teaching the secondary test user;
- The secondary test user, to be instructed about the system by the primary user (where applicable).

We use three main techniques to collect user feedback during practical testing, all complemented with audio documentation for backup and/or notes made by the test leader:

4.3.1 Thinking aloud and observation

This technique means that the test users are given assignments they have to perform with YDS, or come with their own agendas. The test leader encourages the users permanently to talk about his/her impressions and actions during the evaluation process. In such a way, the mental models by which users address a task or try to achieve a goal can be detected and analysed. All the while, the test leader observes carefully the subjects' behaviour in order to try and detect even semi-conscious interactions with the system, or barriers that are not expressly addressed by the user.

Test participants are allowed to digress from the test scenario and perform unplanned tasks. Complementing the controlled interaction with the prototype with unplanned actions provides information about how potential users would use the system in natural settings and may lead to the identification of issues unanticipated in the evaluation script. The benefit of this approach is the fact that user behaviour and user satisfaction become immediately transparent. The need for modifications – if any – will become apparent, as will the possible need for specific training or introduction to the YDS system. At the same time, the professional users will express to what extent YDS actually caters to their everyday work requirements.

4.3.2 Constructive interaction (teaching back)

This technique, which only applies to external evaluators, consists of two stages. In the first phase, one test user gets the opportunity to try out and become familiar with YDS. In the second phase, the same user explains the functionality of the system to the next user in line. The success rate of this direct user-to-user training is directly related to the mutual understanding of the system.

The particular benefit of this approach is that the first user is required to expressly verbalise his/her comprehension of how the system is working and how it is intended to be used. This task therefore triggers a reflection process and prompts the first user to explain YDS in a systematic fashion. This reveals how deep the actual understanding has become at this point and highlights features that remain

unclear or hard to grasp. In case this “Chinese whispers” test works well, the system has a very clear and easy usability; if not, any misapprehensions highlight urgent action points.

However, this approach can only be used in situations when test users have some extra time on their hands and are not anxious to rush back to their regular tasks. It is also of limited use with the user interface simulation and the first prototype, as user interaction with the system requires concise stewardship through the available processes, which users cannot know at this stage.

4.3.3 Collection of express feedback

Immediately after finishing their hands-on experience with YDS, the test users are asked for their personal evaluation of the system. They are asked to fill in standardised questionnaires and are also given the opportunity to independently express their opinion and possible suggestions.

The benefits of this technique are obvious, since it allows the collection of conscious cognitive reactions and recommendations as well as quantitative metrics related to the KPIs (see section 3.5 above). While such information alone, without the abovementioned first two steps, would run the risk of misrepresenting the user experience – e.g., since people tend to rationalise or to respond according to pre-existing prejudices – in this case it constitutes a useful supplement to the observations made during the practical work with YDS.

However, all user evaluations must take into account that users frequently tend to react adversely and insecure to new, unaccustomed software. This is particularly true for those professional users who have long-term experience with other software solutions in the particular field of YDS. Furthermore, the phenomenon that assessments made in surveys frequently turn out more negative than is warranted by the actual subject of inquiry will most probably come up during the YDS user evaluation as well. The situation of being asked for opinion statements as such often leads to a particularly critical point of view.

4.3.4 Questionnaire design

As mentioned above, the questionnaires that will be handed out to both pilot participants and external evaluators (on paper or, preferably, in electronic form) serve the primary purpose of collecting quantitative, respectively quantifiable, information related to the KPIs. To this end, all questionnaires will contain a set of identical questions and scales irrespective of the target group or the stage of development in which they are applied. In such a way, we will be able to perform a differential analysis and detect tendencies in user appreciation over time.

Users may rank each of the standard questions on a five-step scale:

Excellent – good – fair – poor – fail

The scale’s labels may be modified depending on the question, but the gist and left-to-right, good-to-bad direction of the scale will always remain the same.

However, it is not likely that only one question will pertain to each KPI. Instead, we expect to have short clusters of questions per KPI, from which we will calculate an arithmetic average. For instance, it would not be optimal just to ask “How do you rate the overall usability of YDS?” More pertinent would be several questions drilling into specific aspects of the system, such as

- “How easy to use did you find the search function?”
- “How easy to use did you find the function to merge data from different sources?”

- "How easy to use do you rate the creation of graphical charts?", etc.

Additional open and closed questions will be inserted into the questionnaires on demand and in response to specific qualities of the various prototypes. They will be developed in detail once the first integrated prototype emerges in order to reflect its qualities and specifics in an appropriate fashion.

The questionnaires will also record the relevant demographics of the test users (i.e., to which target group they belong, whether they are involved in the pilots or not, what their current professional title is).

4.4 Survey of exploitation opportunities

Having actual potential users try the YDS system offers one of the best conceivable opportunities to find out about the practical application prospects in the media business sector for the eventual product. While participants of the user evaluation efforts will be primarily everyday users and not high-ranking decision makers, it is the assessment of exactly this kind of users that will influence the evaluation process in media companies to a great extent.

If the test users realize the gains in time and convenience and do not show adverse reactions to the YDS system, they might become YDS "ambassadors" in their respective organisations. Under the assumption that the participants in the user evaluation are representative of their peers in similar parts of the sector, their statements can also be used as a supporting measure in the eventual YDS sales development.

4.5 Ascertainment of non-user related KPIs

Some of the non-user related KPIs as listed in particular in section 3.5.3 above can be measured in a quantitative fashion: The time it takes a user to complete a task, the number of social/semantic links established, and the number of third-party organisations who have taken to using YDS.

Other functional indicators must be determined by human experts, such as the validity and usefulness of social enhancements of YDS data and the system's capabilities in the areas of queries and overarching ontologies. In the former case, we will most likely ask users from the pilots to classify the individual social enhancements, and then calculate the ratios. In the latter case, YDS project experts will demonstrate the achievements manually and by example within parameters that will be developed in the course of the project.

4.6 Risk assessment

As described under WP1 in D1.1 Project Quality and Assessment Plan, there are possible risks that have been outlined in the context of the pilots' operation plan, and which may also affect evaluation. These risks along with the proposed solution and mitigation strategies are described below:

Problem/Risk	Actual/Potential Impact on the project	Risk Exposure	Proposed solution/mitigation strategy
R6.1 Possible difficulty	If the number of pilot users is too	Medium	Pilot users will be selected not only based on their competences, but also on their

Problem/Risk	Actual/Potential Impact on the project	Risk Exposure	Proposed solution/mitigation strategy
identifying and recruiting fully committed pilot users	low, or if pilot users change too frequently during the course of the project, their feedback may be inconsistent and therefore not useful as input for the developers		intrinsic interest in the services to be delivered by YDS, i.e., their regular work will benefit from taking part in the pilots. Moreover, YDS will support pilot users in accessing third party funding where applicable (such as grants). And finally, several user partners have permanent staff members on hand who are qualified to take part, which will help reduce the reliance on third parties.
R6.2 Participating in pilots may frustrate users due to possible shortcomings of the prototypes or lack in functionality	If users are dissatisfied with the YDS prototypes and do not recognize sufficient benefits from using them, they might either abandon their participation or provide unfavorable or unproductive feedback	Medium	Training and tutoring will be provided, and pilot participants will be proactively shepherded throughout the entire process. This is to enhance their understanding of the system as such, and to mitigate the obvious shortcomings of "beta versions". Moreover, many pilot participants will be involved in the definition of user requirements, which renders their dissatisfaction with the prototypes less likely.
R6.3 Way more effort required than planned to manage pilot projects and their users effectively	If the effort to shepherd the pilot users through the pilot activities overstretches the capacity of the user partners, results and feedback may be insufficient	Medium	There is a dedicated task on deployment planning (T6.1) which will allow the WP leader as well as the pilot operation partners to define and assess their input at an early stage. One element of this planning is to set up the pilots in such a way that they remain low- to medium-maintenance. If necessary, they can therefore be adapted or modified before they have started to run. Another mitigating factor is that several user partners will use their own staff to operate parts of the pilots and thus have good management capacities and direct communication channels already in place.

Table 10: Risk Assessment for Pilots Deployment and Evaluation[25]

[25] YDS Deliverable 1.1, p. 49-51 (slightly edited)

The above risks will be monitored closely during the entire evaluation process and the appropriate mitigation measures takes if and as appropriate.

4.7 Ethical considerations

The YDS user evaluation methodology presented in this document requires the collection of personal data, such as interaction data with the system, demographic data and responses to questionnaires. In order to conform to the privacy rules and the national and EU legislation we suggest that all users participating in the evaluation process give written consent before taking part in the pilots and evaluation procedure.

To this end, we will provide all users with written information in understandable language and appropriate format. This will include what the project is about, who is participating in it, what are the expected outcomes, how they will be communicated, as well as which and how the personal data of user participants will be stored. The privacy of all participants will be respected and all effort will be made to keep their personal data confidential. Furthermore, participants shall have the right to refuse to take part or withdraw from the evaluation procedure at any point.

4.8 Time plan

The time plan for monitoring and evaluation of YDS is determined by the due dates of the deliverables. Obviously, the plan can only be respected if the technological development of YDS reaches the necessary stages in good time. Here is an overview of the time plan, deliverables, and responsibilities.

Activity	Due date	Responsible
Evaluation methodology	M7	EJC
Pilot planning	M9	DW
Launching pilots	M10	EJC, DW, GFOSS, MAREG, D/PER, NUIG
Testing of the first integrated prototype	M13	EJC, DW, GFOSS, MAREG, D/PER, NUIG, NCSR/D, ATC, TF
Intermediate pilot monitoring and evaluation report	M24	EJC
Testing of the intermediate integrated prototype	M25	EJC, DW, GFOSS, MAREG, D/PER, NUIG, NCSR/D, ATC, TF
Testing of the final prototype	M34	EJC, DW, GFOSS, MAREG, D/PER, NUIG, NCSR/D, ATC, TF
Final pilot monitoring and evaluation report	M36	EJC

Table 11: Evaluation time plan

5 Summary

In all, YDS intends to work with at least about 10 users in each of the three pilots, plus at least 5 external test users per pilot, bringing the overall tally to a minimum of circa 45 test persons to be involved in each of the three rounds of evaluation (i.e., around 135 responses in total). Based on the discussion above, this number will unearth the vast majority of issues and other relevant user observations.

Users belonging to the relevant target groups will be recruited through the respective user partners of the Consortium, which represent civil society, government, and journalism.

We will implement a permanent, informal user evaluation through the active pilots and run dedicated evaluations on the three prototypes. In this set-up, the pilots will be employed to support the co-operative development of YDS and to establish real-world baselines, while the dedicated rounds of evaluation will gauge the actual success of the project and identify the most relevant amendments and further developments required after the release of the first two prototypes, respectively.