UDC 37.014.5:004

FUNCTIONALITY OF THE KSU FEEDBACK 3.0

Alexander Spivakovsky, Dmitry Berezovsky, Sergey Tityenok
Kherson State University

Evolution of e-service "KSU Feedback", analyzing the flaws and limitations of the old version of "KSU Feedback 2.0 RC1", review of functionality and a key features of the upcoming release "KSU Feedback 3.0".

Keywords: *Feedback Service, building circuit of feedback, survey of the target groups, development, poll*

Introduction

KSU Feedback service was developed to provide adequate means for building circuit of feedback. In other words it is the system for gathering and analyzing data taken from interview of anonymous respondents. One of the main goal here is to reach full anonymity of requested groups, and distinctly determine these target groups.

This service is web-based and has a multi-tier architecture consisting of complex presentation tier, data processing and security tiers. Besides, it has useful means for tracking the process of voting and mechanism for storing the data in arbitrary structure.

After each approbation of "KSU Feedback" we have always faced with some system unsuitability or lack of functionality, however they have fully meet their current requirements. It was not a big surprise, as it is important to keep in mind that this sector of software was not much investigated yet. So, we obtained an experience by making our own mistakes.

In this article we propose you to track the hard way we made from the first requirements to requirements of version 3.0. By walking this way once again we will possible see what is wrong and how it should be. Therefore the main goal is by describing the fundamental principles of new version of "KSU Feedback" declare its functionality and design new architecture. Ideally new architecture should cover not only the needs we have now but also should have an ability of easy extension for further requirements.

Long Way to Version 3.0

The first work on this direction was started at the end of 2008. The project called "KSU Students" was supposed to be an alternative to the popular that time site "Univerlife"(http://univerlife.com/). That site was designed to collect anonymous text comments from students about theirs professors. The killer feature of the KSU feedback, comparing to the university life, was to be a way to express an opinion - instead of text comments. We asked students to answer some questions and fill theirs answers in the form. These questions was allow to evaluate some of the professional skills of the professors, definitely just subjective opinion of the students.

By the beginning of 2009 the prototype was written on PHP. Key features was:

- An interface for adding and editing information about the professor;
- For organizing datum was used tree structure that represented infrastructure of the university (university - faculty - cathedra);
- Editor for surveys;

The site was hosted on free PHP hosting and was available as http://students.ks.ua/. The database was filled with information about professors and assistants of the cathedra of computer science of the Kherson State University. After that a test survey was held in one of the groups.

However as results we clearly understood that system is not flexible, extensible robust and user friendly.

This brunch of development has finished its existence and work has begun on the next one.

KSU Feedback and KSU Life

In autumn 2009 we started to work on the new portal. The main goal was to use previous experience to create system devoid of down sides of the KSU Students. We created a list of the new important requirements:

- User should be able to create a poll by himself, ie be able to compose and reorder the questions;
- The questions should be different types: Scale questions, "Select one from the list of predefined answers...", "Select one of many answers from the predefined list";
- One poll can be used for many surveys;
- Results should be presented as histogram for scale questions and as pie chart for the questions with predefined answers;
- Vote anonymously - to be able to vote student should obtain a secret key which allows to take a part of the single survey.

We selected Python 2.6 and Django framework as a development platform for the new system. It should be noted that application was spitted on two independent parts - the first one is interface for vote, and the second one - administration module for creating polls, surveys, and displaying results. Besides, the first module was responsible for authorization of the respondents, key generation and voting.

As communication layer was used simple ourselves-designed protocol that works over HTTP.

First release was launched winter 2009 and was immediately tested on several groups of students.

But as soon as number of conducted interviews started to grow, it became clear that the system of the data storage needs change, and the fact that there is no any security system makes impossible any type of the collaborative work on the voting organization.

KSU Feedback 1.0

Experience has shown that two separate modules bring a lot of inconvenience in the management of the system. This makes process of creating and conducting the survey extremely long and annoying. Therefore we decided to merge two parts in one.

This version was released in the middle of the 2009 and has been used in the next six month by the Department of Computer Science of the Kherson State University.

KSU Feedback 2.0-alpha

In early 2010 on the basis of the experience of using previous KSU Feedback we were started to develop a new version. This version was based on the source code of the previous one in its entirety, but the new version contained a number of important changes:

- Introduced the concept of the branches(nodes) which was a sort of the directories in the file system. Each entity(poll, report, survey) was linked to the tree. This streamlined the huge amount of accumulated data, sort and organized them. We tried to create a GUI similar to MS Windows Explorer well known for office employers to simplify theirs work;
- We added the security system based on the concept of the Unix file system security - each entity was kept at a level of information access:
- Owner's;
- Group;
- Other users.
- We created automatic visualizer of the results (in the old version to see the results it was necessary to create a report - a separate entity).

This version has replaced the existing and has been used for some time. However, work on a version beta did not stop.

KSU Feedback 2.0-beta1, KSU Feedback 2.0-beta2

The beta version included a number of improvements, which mainly related to the alpha version bug fixes, as well as expanded security system. The new security system has become possible to assign permissions to individual users and user groups. User groups were considered some users attached to the same home directory.

KSU Feedback 2.0-RC1

This version is the latest in the 2.0 branch. It was launched autumn 2010. Important changes are:

- Support for widgets for results visualization;
- API for integration with third party services;
- One key set can be attached to many surveys, it greatly simplifies survey managing.

Historical Analysis of the Results

The last production version of "KSU Feedback" was "KSU Feedback 2.0-RC1". This version is the most interesting for our analysis from two perspectives:

- It is the latest version
- It has a lot of modifications after it actual release. That means, that we have added new features on existing code basis.

After each release we made some testing and also discovered bugs during exploitation. We added bugs to our development bug tracking system.

When we added a new functionality to this version, we made the same procedures.

We noticed that with each additional features number of bugs in track is growing.

We think that main reason of it is that initial code was not targeted for extending to features we was needed.

Each new part of functionality can be contingently estimated in percentage ratio of old functionality, depend on time our team has spent on it implementation. Below is an approximate graphic that demonstrates the relationship between percentage of functionality implemented in number of issues in a bug tracking system

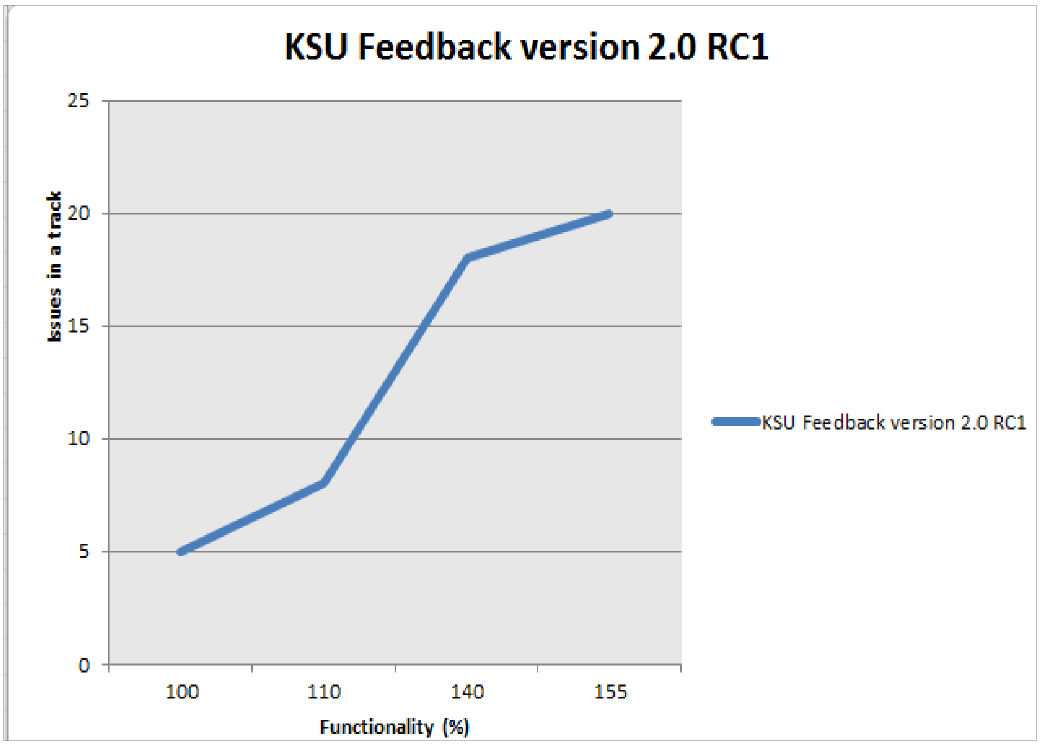


Fig. 1. Bugs in the bug tracker

As you can see, graphic shows not optimistic outlook for any additional functionality. Most likely we will get very unstable software if we will try to support and grow existing code basis.

Experience is the Mother of Wisdom

At this point it is clear that we need to completely rewrite the service. And therefore we are free to choose a new stack of technologies. PHP and Python with Django was very comfortable for usage at the begging of the project but with further growing this stack became harder and harder for extending and refactoring. So, we will try to move from the traditional LAMP to Java technologies stack, that promises to be much more easy in usage in big projects. Unfortunately scope of this article doesn't touch topic of the technologies so much, therefore we need to left this not fully described and move to another points.

Of course, we completely aware of problems connected with full rewriting of application on the new stack: we will need to re-write not only the parts, we are not satisfied with, but also the components which worked pretty much good. At first look it is bad, but we have no any other alternative if we want to have quality and comfortable soft at the end. In other case we are risking to invest much time for implementing any other additional feature that in addition may ruin the work of older components and bring a new portion of bugs.

So lets consider that previous versions was only a prototypes for investigating final requirements for the "KSU Feedback 3.0". And they gave us the main - user experience, that we can analyze and enhance in the next versions. Below we will disassemble key components of "KSU Feedback" service we have now, and think how we can change this to make user experience better.

Poll Creation

Comparison of the old and new version given in Table №1.

Table №1.

Actions needed to create a poll

Version	Description
KSU Feedback 2.02	To create a poll user needs to select node for storage, set name. New object will be created. Than go to edit mode and add questions one by one with edit form.
KSU Feedback 3.0	Click "Add Poll" button. Poll will be created in the user personal zone with default name. User will automatically redirected to edit mode. Creation of the poll will mostly like writing the list of the questions in any common rich text editor.

What has been changed?

- User, who is creating a poll doesn't need to think where to place it. This can be done any other time or by person who responsible for this;
- Less actions for doing more: user have not to click buttons for adding, editing, deleting question or appending the variants. He just needs to type and specify the question type.

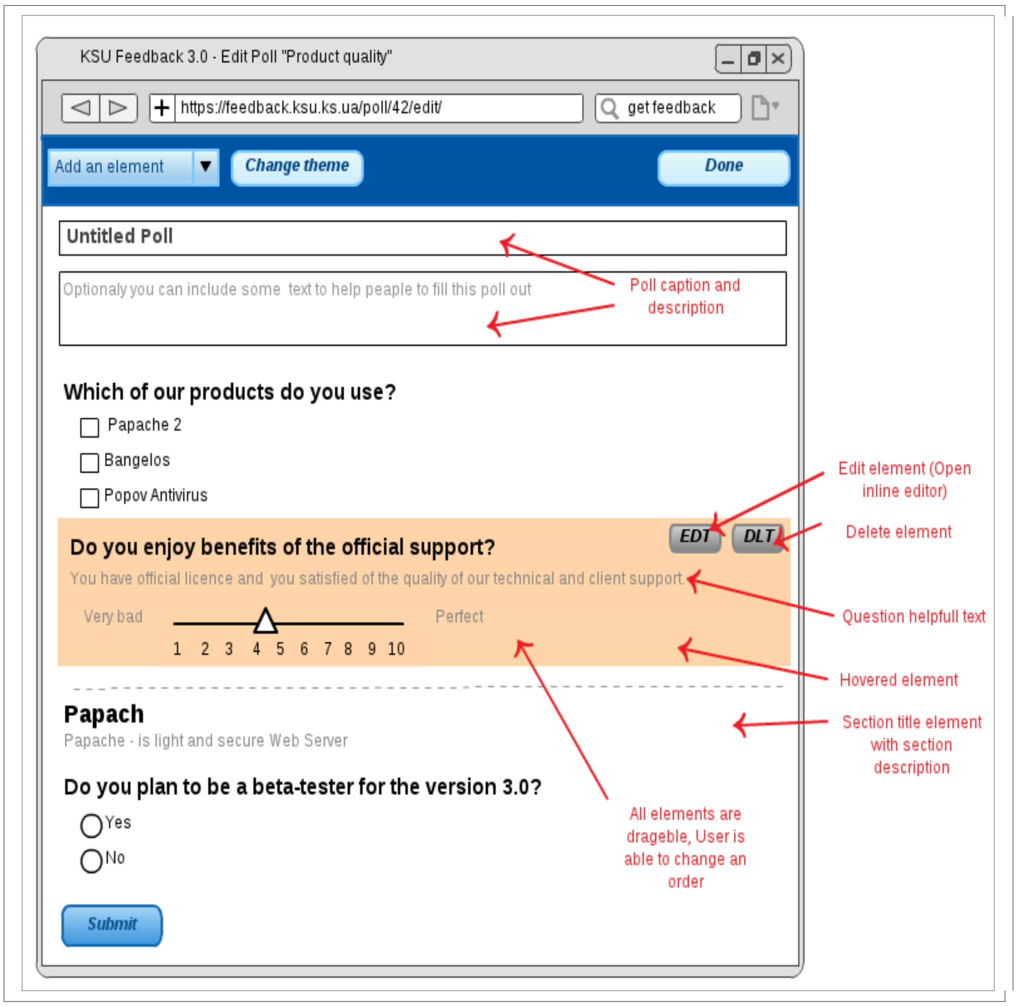

KSU Feedback 3.0 - Edit Poll "Product quality"
https://feedback.ksu.ks.ua/poll/42/edit/
get feedback
Add an element
Change theme
Done
Untitled Poll
Optionaly you can include some text to help peaple to fill this poll out
Poll caption and description
Which of our products do you use?
Papache 2
Bangelos
Popov Antivirus
Edit element (Open inline editor)
Delete element
Do you enjoy benefits of the official support?
EDT
DLT
You have official licence and you satisfied of the quality of our technical and client support.
Question helpfull text
Very bad
Perfect
1 2 3 4 5 6 7 8 9 10
Hovered element
Papach
Papache - is light and secure Web Server
Section title element with section description
Do you plan to be a beta-tester for the version 3.0?
Yes
No
All elements are drageble, User is able to change an order
Submit

Fig. 2. Mockup of the page for editing poll

Organization of the New Vote

Comparison of the old and new version given in Table №2.

Table №2.

Actions needed to organize a new vote

Version	Description
KSU Feedback 2.02	Select place in hierarchy for creation. Specify poll, set name. Append key sets. Give cards with access code to each respondent.
KSU Feedback 3.0	Click on the poll and click “Make survey...”. Set name. Provide some meta description of this survey, that can be use for sorting in future. For example entity name, category, department, etc. Create/select target groups. **Target group** - is rules of access which defines any group of respondents. In other words, it defines the way of access for voting. That means it can be voting with secret key on a cards, special link in email, open voting, voting in limited time interval or even entry to defined group in some of the social networks.

What has been changed?

- Easy way of creation of new survey object;
- Surveys have properties, so it can be grouped or categorized;

- Different types of access to voting;
- Approach with using of Target groups allows to define some target group once, and use it in a future.

Voting Process
Comparison of the old and new version given in Table №3.

Table №3.

How does the voting process look like

Version	Description
KSU Feedback 2.02	Organizer was needed to print keys for each voting. Voting can consist of several surveys. Then, respondents somehow receive keys. The next step is to go to “KSU Feedback”, log-in using security key, respond on the surveys.
KSU Feedback 3.0	Respondents actions depend on what type of access mode was selected for the survey. It varies from voting without log-in to described above authorization by security key. To vote, respondents must go to special company URL that consists of KSU Feedback URL and sub-domain that is a company name. For example: mybigcompany.feedback.ksu.ks.ua

What has been changed?

- Different types of access to voting have been added
- Each company has their sub-domain
- As each company has their own page on a site, users will have more content to look without registration.

Tracking the Status of Voting
Comparison of the old and new version given in Table №4.

Table №4.

Means for tracking the voting

Version	Description
KSU Feedback 2.02	User can get current results of any survey by selecting such item in context menu. But he need to find this item in hierarchy. And this task become harder and harder as company objects scope is growing.
KSU Feedback 3.0	There is special page that displays the current voting. By default items are grouped by the poll they are referred to, but user can group them by meta keywords, status or just use quick search on this page. It is possible to see quick results for all selected surveys.

What has been changed?

- User is able to see all surveys which are now participating in the voting;
- User can easily find any needed survey using search and grouping;
- User can see quick result not for one poll but for the set of polls.

Browsing the Results
Comparison of the old and new version given in Table №5.

Table №5.

Means for browsing the results

Version	Description
KSU Feedback 2.02	There are several options of browsing the results: quick results, reports, queries, using the API. There no abilities of exporting of the results.
KSU Feedback 3.0	Has only three ways of browsing results: quick results/results, queries and API. As you can see there is no report item. We noticed that users anyway don't use built-in text editor in our service. Therefore, we decided to provide them with great abilities of exporting results to PDF and PNG, so they will free to use their favorite text editor for composing the report.

What has been changed?

- Users are free to use any text editor;
- Enhancement query editor.

Security

Comparison of the old and new version given in Table №6.

Table №6.

Permissions distribution

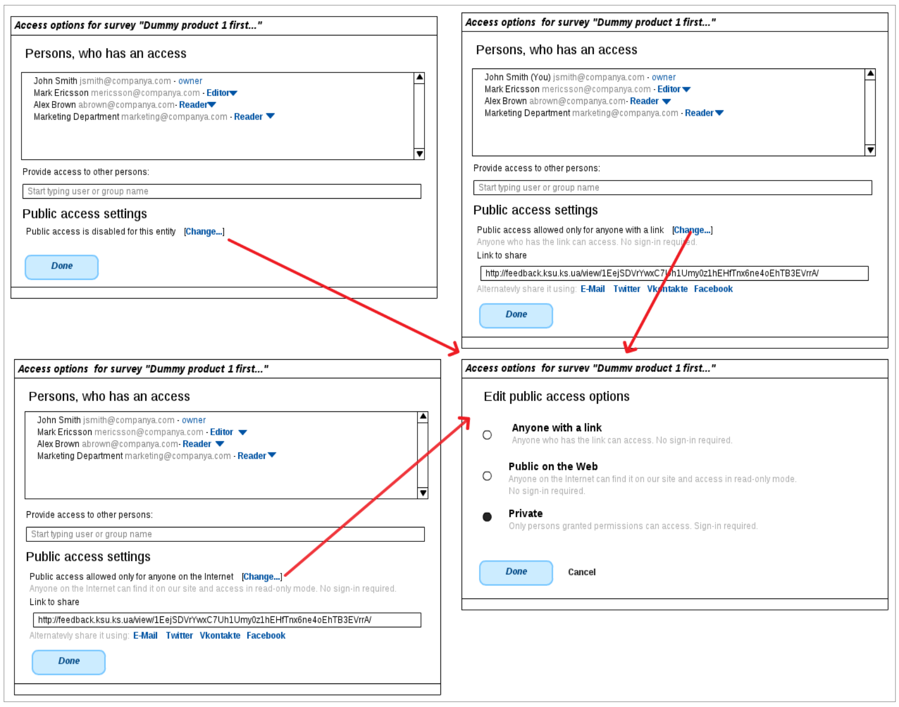
Version	Description
KSU Feedback 2.02	We have a hybrid of well known approaches - Unix like permissions (legacy of the one of the earlier versions) and ACL's that were implemented in RC1. We have a lots of problems in this subsystem starting from very unusable and annoying UI to critical bugs on the server side. Definitely we need completely new redesigned security system. Here outlined only a couple of problems: • We support user groups but users are grouped by home directory that is only one per user; • It is extremely hard to understand for the and user what does "read access for survey Some Survey" means; • GUI for editing access is not usable and buggy.
KSU Feedback 3.0	Users are attached to the company. And has identifier that contains both company and user name, eg. jhon@somecompany.com. This means that login jhon should be unique only in "somecompany". Users can be grouped manually for easy access distribution. Group identifiers are similar to user identifiers and looks like <group name>@<company name> Groups can consists unlimited range of users but cannot include another groups. Instead of old access levels we present the following: • <u>none</u> - access is not allowed, • <u>read only</u> - access only for view, without edit. • <u>edit</u> - access with edit permissions, • <u>full</u> - full access includes grand permissions, this means that user is able to change permissions on this entity. Public access is a separated property of the access level. Public access means access without authorization: • <u>none</u> - public access is not allowed, • <u>link</u> - access is allowed for users that have direct link for the entity, however this entity isn't listed into the catalogs of the portal or any charts and ratings,

	• public - the same as previous but this entity will be listed in the catalog of the public entities on the site and can appers in the charts and ratings. The important restriction for public access: all public access levels except of none allows read only access.

What has been changed?

- Removed old Unix based approach;
- Used modern and flexible ACL;
- Regardless on using ACLs user is know nothing about it. There are just a couple access settings that are clean even without description.

The UI mockup below illustrates the GUI for settings up access for an entity:



Access options for survey "Dummy product 1 first..."

Persons, who has an access

John Smith jsmith@companya.com - owner
Mark Ericsson mericsson@companya.com - Editor▼
Alex Brown abrown@companya.com- Reader▼
Marketing Department marketing@companya.com - Reader ▼

Provide access to other persons:

Start typing user or group name

Public access settings

Public access is disabled for this entity [Change...]

Done

Access options for survey "Dummy product 1 first..."

Persons, who has an access

John Smith (You) jsmith@companya.com - owner
Mark Ericsson mericsson@companya.com - Editor▼
Alex Brown abrown@companya.com- Reader ▼
Marketing Department marketing@companya.com - Reader▼

Provide access to other persons:

Start typing user or group name

Public access settings

Public access allowed only for anyone with a link [Change...]
Anyone who has the link can access. No sign-in required.

Link to share

http://feedback.ksu.ks.ua/view/1EejSDVrYwxC7Uh1Umy0z1hEHfTnx6ne4oEhTB3EVrrA/

Alternatevly share it using: E-Mail Twitter Vkontakte Facebook

Done

Access options for survey "Dummy product 1 first..."

Persons, who has an access

John Smith jsmith@companya.com - owner
Mark Ericsson mericsson@companya.com - Editor ▼
Alex Brown abrown@companya.com- Reader ▼
Marketing Department marketing@companya.com - Reader▼

Provide access to other persons:

Start typing user or group name

Public access settings

Public access allowed only for anyone on the Internet [Change...]
Anyone on the Internet can find it on our site and access in read-only mode. No sign-in required.

Link to share

http://feedback.ksu.ks.ua/view/1EejSDVrYwxC7Uh1Umy0z1hEHfTnx6ne4oEhTB3EVrrA/

Alternatevly share it using: E-Mail Twitter Vkontakte Facebook

Done

Access options for survey "Dummy product 1 first..."

Edit public access options

○ **Anyone with a link**
Anyone who has the link can access. No sign-in required.

○ **Public on the Web**
Anyone on the Internet can find it on our site and access in read-only mode.
No sign-in required.

● **Private**
Only persons granted permissions can access. Sign-in required.

Done Cancel

Fig. 3. Mockups of the Access Level dialog

Navigation

Comparison of the old and new version given in Table №7.

Table №7.

Navigation in user structure

Version	Description
KSU Feedback 2.02	User navigates in tree structure and has an ability to search objects by name in each selected node. Also it is possible to group objects by type or keep them ungrouped. Each object belongs to only one node.
KSU Feedback 3.0	Each object are marked with a label. Labels themselves are united into tree structure. Objects can be grouped by labels, or by meta keywords, or by date. Global search is also available.

What has been changed?

- New approach of viewing user objects;
- Quick search is now available;
- Unused functionality have been removed.

Companies Separation

Comparison of the old and new version given in Table №8.

Table №8.

How one companies on KSU Feedback differs from another

Version	Description
KSU Feedback 2.02	Each company is the first child of root element
KSU Feedback 3.0	Companies don't belong to some single root, they are not connect with each other. Each company has their own sub-domain. For example *micronoft*.feedback.ksu.ks.ua, where micronoft is a name of a company.

What has been changed?

- Companies are not united into one structure, so they are completely independent from each other;
- Each company has a comfortable URL for log on.

API

Comparison of the old and new version given in Table №9.

Table №9.

Provided API

Version	Description
KSU Feedback 2.02	There were 3 types of external API: 1. Data Access API - allows to fetch datum in JSON format. In the RC1 was available to fetch survey results, poll info and some other; 2. Authorization API - allows registered users of the third-party site to vote on the KSU Feedback without the key. Authorization process in this case is delegated to the third-party side; 3. Widgets - easy-to-use widgets for displaying results of the survey, or a single question of the survey. Was available 2 widgets in RC1: Histogram widget, and pie diagram.
KSU Feedback 3.0	We plan to create XML and SOAP API for all operations. But not all of them will be available over the web. The following API will be accessible: ● Poll management; ● Survey management; ● Data API. Widgets will be still available

What has been changed?

- Standardized format;
- Big range of abilities;

UI

Comparison of the old and new version given in Table №10.

Table №10.

UI Paradigm

Version	Description
KSU Feedback 2.02	Managing should be similar to the managing of the file system.
KSU Feedback 3.0	We still have object containers (Collections) but UI is focused on easy access and management of the objects. Tags and meta information will allow us to display to user the most relevant results for the query.

What has been changed?

- Completely new concept.

Summary

In sections above we gave the overview of past versions of "KSU Feedback" and made short description of what functionality the next version should have to meet the current needs. We propose to make these changes from the perspective of the experience of using live system we have now.

The version 3.0 will be much more suitable than its precursor and it will be a big step forward. But it also has a back side. We will try to give a short review for each aggravating aspect and unresolved problem.

First of all, we need to consider that even if we will cover all existing needs, we are not able to anticipate what will be needed in a future. This is naturally as "KSU Feedback" is very young system and it has not fully defined the scope of its application yet. We noticed that, each new extending of usage scope usually invokes significant changes in the code, to cover the tasks of the new area. At this point it is not clear in what direction we will need to make the next step, but one is known for sure - "KSU Feedback" is becoming more and more global system.

Also, let's don't forget about users, which will need to face with completely new UI and business logic. Some of them has already used to know our old bugs and issues and then will need to get familiar with the new one. Don't think the authors of this article are pessimists, but problems are always exist in software and good proof for that is so called "Black swans theory".

Another bad side of this is unsupported old version, as for reasons which are out of these article, developer team are very limited in resources, so there will be no time for fixing issues in old version.

But anyway, we believe that creation of "KSU Feedback 3.0", that will follow all described concepts has more positive sides than negative ones. Users will get easy, comfortable tool for creating the circuit of feedback. System will requires minimum of users time to produce maximal results. And the main advantage of the new version, is that we will have service with room for growing. In other words we will have clean way for reaching new goals.

REFERENCES

1. DeMarco T. Productive Projects and Teams / DeMarco T. — New York: Dorset House, 1999 — 245p.
2. Cohn M. User Stories Applied For Agile Software Development /Cohn M. — Boston: Addison-Wesley Professional, 2004 — 304p.
3. Wiegers K. Software Requirements / Wiegers K. — San Francisco: Microsoft Press, 2003 — 544p.
4. Kierevsky J. Refactoring To Patterns / Kierevsky J. — Boston: Addison-Wesley Professional, 2004 — 400p.
5. Tognazzini B. Tog On Interface / Tognazzini B. — Boston: Addison-Wesley Professional, 1992 — 352p .
6. Rogers Y. Interaction Design / Rogers Y. — Malden: John Wiley & Sons, 2011 — 602p.