Strategy 2017 - Review

After our initial forming as TB in 2016, we decided to create a paper that outlines a general strategy with respect to the ILIAS Vision 2015. The idea was, that we would need some general direction in which we would want to move ILIAS. This would allow us to adopt and discard tasks in a coherent way, focus on matters we deemed important and not get lost in day to day duties and manifold requests from the community.

The paper[1] we created is partitioned in five sections that each focus on one aspect we wanted to improve. Each section in turn contains more or less measureable goals that substantiate the general direction given as a short statement on top of the section. This review will be structured accordingly.

After our first (complete) term in office, we want to review what the community and we have achieved so far regarding these aspects and goals. For one, this is a self-review of our activities which might help the community to judge whether it wants to trust us for another term in office. On the other hand, a new TB will be elected and even if some persons of this TB might be members again, the next incarnation of the TB should be able to reevaluate or even completely recreate a strategy based on our experience, failures and achievements.

Community Driven

We see that the community has made a huge leap forward in this aspect of our strategy. We made progress on almost all goals in this section. Some of the improvements we see today are so big, that we wouldn't have dared to dream of them two years ago. Most notably we see these improvements:

- The amount of visible contributions and contributors was raised, at least regarding code in our repository. As of today, we have seen more than 1.700 PRs on GitHub, some of them by people never seen before. We even have converted one contributor to a maintainer.

1https://docu.ilias.de/goto_docu_file_5585_download.html

- The quality of the contributions improved, as witnessed by new READMEs, Testcases and Unit Tests, a new format for feature requests, the decreasing count of dicto violations and the general level of our discussions and arguments, regarding code as well as features.
- The Contributor Model as a new way to maintain code collaboratively works out for the UI-framework and will be used for other components in ILIAS.
- Technical means of backing our processes gained traction. The results of the CI-Server are a regular topic on the JF, using Travis as CI server made failing unit tests visible on GitHub.

Although we see substantial improvements here, there are still areas regarding the community that leave room for further improvement:

- There still is code in our repository that is more or less unmaintained. The term "implicit maintainership" was introduced for people that somehow take care of such code without seeing themselves as real maintainers. This made this code more visible. It still isn't clear how such code can be moved under proper maintenance.
- Although we gained a lot of contributions it is still hard to recruit new maintainers for core components of ILIAS. Users with special needs are not included in our development process on a regular base, although the recently found "SIG Barrierefreiheit" (= SIG Accessibility) sparks hope that this could change soon.
- There is a lot of room for improvements regarding our technical tools that back our processes. A lot of areas still involve manual labor that should well be automatable, e.g. compiling CI-results for the JF or packing a new release for ILIAS.
- The current focus on open bugs as a quality measure seems to be too narrow. The discussion about what quality work means in our community should be widen to a maturity model for components that takes more criterions into account.

We thank the community for the collaboration regarding our efforts in this area. Changing habits can be hard sometimes. We are looking forward to even more growth in our community, regarding quality as well as quantity.

Reliably Learning Management

We see somehow mixed results in this area. There are aspects where we surely made progress here, but we also see aspects that almost didn't change at all or where the goals even are questionable in the first place.

We see some improvements on security related aspects of reliability. We introduced `security@lists.ilias.de` as a channel to inform about security issues and managed to quickly answer the incoming mails and resolve the issues raised in them. There is work to be done here, e.g. strengthening the staff and processes regarding that channel, but the general direction here seems to be correct. There are issues with the opposite way of communication, i.e. how and when the community is informed once a security issue was found and resolved. We attempt to find volunteers to operate the security-list, that, once recruited, could also handle the outbound communication.

Regarding training of our developers and building awareness for security related aspects of software development we certainly made some progress, e.g. by having workshops regarding security. We are looking forward to introduce a new tool to harden ILIAS regarding input to the system that currently is under development. A call for bids to extend the documentation how to securely configure ILIAS currently is open.

Unfortunately, we see that we didn't improve regarding the performance aspect of reliability. The performance measurement tools we have do not work reliably or continuously and do not allow to conclusively evaluate performance changes of the system. Although a switch of the person in charge for the tool itself was done successfully (thanks to both!) and new JMeter-scenarios have been created, it still is hard to transform the project initiated by the SIG Performance into a permanent arrangement.

The tooling with regards to reliability have improved on the unit test front, on the other hand. Unit tests now are a requirement for the `./src`-folder, the number of available unit tests shows a constant growth. We are looking to revise our goal regarding automated testing for security issues, as we came to see the approach to offer tools that nudge or force developers to develope secure code as more promising.

The goal, that ILIAS is installable and upgradeable at any point without concerns

on performance, security and reliability, seems to set a very high, maybe even insurmountable, bar. We are looking to replace this goal by more concrete and reachable goals in our next iteration of the strategy.

Due to external circumstances and driven by the community, the security aspect of reliability had a strong focus in this section of the strategy in our last term. We think we can be contended how we all handled the challenges here, although we will require further improvements. For the next term we will have the challenge to transform the operational task in security to a more permanent arrangement so that we can take care of other reliability aspects more.

Usable for Everyone

For the goals outlined in this section we mostly see positive results and even some large improvements, although we need to address some goals more effectively.

Most notably we see improvements in the concepts and considerations of UI related aspects in feature requests. These are a fixed part of every feature request discussed in the Jour Fixe now, which lets us address UI-related questions early in the process. The introduction and increasing usage of the UI-Framework/ Kitchen-Sink helps concepters to name, find and reuse existing components as well as developers to implement these UI-components in their component. There still is some work to do regarding the usage of that tool by concepters, though.

The increasing adoption of the UI-framework will also be a good base to work on accessibility issues in the future, which only had a very minor focus in our last term. We also did not succeed in rationalizing UI decisions by using data generated by UI-tests performed with different user groups. Although some single institutions performed such tests, we are not there yet when it comes to systematically using them in our decision-making.

We also see that more stakeholders start to consider multiple perspectives on the system instead of their own requirements only. The Page Layout certainly is an example for a project that included a lot of different usage scenarios from various parties. We also tried to represent the perspective of the overall system in the discussion of the restructuring of the Test and Assessment. We still see, that the singular requirements of stakeholders and the perspective of the overall systems often are at odds and will try to reformulate the goal towards a more actionable

goal.

Learning Everywhere Anytime

We consider our actions in this part of the strategy a complete failure, starting from the formulation of the goals, over their execution to the actual developments in this area. We currently do neither have an understanding of how we see or even implement device independence of offline scenarios in general ILIAS nor did we make progress in architectures or processes that support these scenarios.

We do recognize that community members (the Pegasus Group) managed to build an app for Android and iOS that supports ILIAS, however the app was build outside of ILIAS development processes (for perfectly understandable reasons). Our strategy and actions did therefore not influence its development and thus the result is not something we consider further when evaluating our work. However we believe that collaboration with the community members now experienced on this field will be fruitful in future.

Regarding our general strategy, we will completely review this section. Most likely we will emphasize "connectivity" more, to improve ILIAS' capabilities to interface with apps as well as other external systems. This aligns with the agreement we reached with the Pegasus Group to attempt to collectively work on backend issues like webservice-interfaces or OAuth-authentication to support an app instead of the, more controversial, app itself. We also see that the general handling of client-side code will need more emphasis in our strategy.

One little bright spot can be noted on this section, though: The question how UI-components are displayed on small devices was addressed in the context of the UI-Framework for some components, e.g. in the development of the Page Layout, so we will see little improvements in the usability of ILIAS on these devices.

Beyond Standard

This is an area where we mostly missed the goals we set in our strategy and only made little improvements.

We made a little progress in making concepts more transparent by starting to create READMEs for various areas of the application. The improvements here are small, however, since READMEs and documentation can only play some role in

this endeavour. On the other hand, new concepts have been introduced, which makes the inventory larger.

Regarding the architecture of ILIAS we also made some small progress, e.g. Global Screen/Main Menu, the File Delivery or the components in the src-Folder, which will make it easier to swap, mix and match parts of ILIAS. On the other hand it is often tedious to roll out architectural changes over the complete codebase, as demonstrated by the ongoing introduction of the DI-container.

On the plugin-front we see that we failed to make progress in our current term, we do not see improvements here. This area might be a target for a reevaluation of our goals for the next strategy paper.

We also failed on the introduction of coding and style-conventions for core-code and plugins, although there now is a proposal to introduce PSR-2. Changes like these might be low hanging fruits and technically simple to implement, but also might have severe consequences for the system, e.g. regarding fixes and patches for different branches, and thus could upset various stake holders.

Regarding the customization of skins, we see mixed results when looking into different groups of skin-creators. For casual users it is easier now to switch e.g. colors in the delos-skin. For high-end-users on the other hand there were little improvements.

Conclusion

We think we all can be proud of the improvements that we implemented regarding the software as well as our community in our term of office. We certainly did not achieve every goal we set in our strategy and we even can see now, that some goals might not be what we really want or need. This is fine for us: If we had achieved all goals in the strategy the bar would have been to low or our vision would have been too cautious. One certainly can see clearly, on which areas we spent most of our time, by choice or by external reasons.

We are looking forward to revise the strategy for our next term and to achieve even more with all of you. We kindly ask for your ongoing participation and support of our work, because we will not be able to achieve anything without all of you.

Office:
Ebertplatz 14-16 · D-50668 Koeln
General Manager: Matthias Kunkel
VR 16017 / Amtsgericht Köln

Technical Board:
Timon Amstutz
Richard Klees
Fabian Schmid
Michael Jansen
Alexander Killing

Contact:
E-Mail: tb@lists.ilias.de
Web: www.ilias.de