Metrics Driven Research Collaboration: Focusing on Common Project Goals Continuously

Marc Schreiber*, Bodo Kraft*, Albert Zündorf†
* FH Aachen - University of Applied Sciences
Medical Engineering and Technomathematics
Jülich, Germany
† University of Kassel
Software Engineering Research Group
Kassel, Germany
Email: marc.schreiber@fh-aachen.de, kraft@fh-aachen.de, zuendorf@uni-kassel.de

***Abstract*—Research collaborations provide opportunities for both practitioners and researchers: practitioners need solutions for difficult business challenges and researchers are looking for hard problems to solve and publish. Nevertheless, research collaborations carry the risk that practitioners focus on quick solutions too much and that researchers tackle theoretical problems, resulting in products which do not fulfill the project requirements.**

In this paper we introduce an approach extending the ideas of agile and lean software development. It helps practitioners and researchers keep track of their common research collaboration goal: a scientifically enriched software product which fulfills the needs of the practitioner's business model.

This approach gives first-class status to application-oriented metrics that measure progress and success of a research collaboration continuously. Those metrics are derived from the collaboration requirements and help to focus on a commonly defined goal.

An appropriate tool set evaluates and visualizes those metrics with minimal effort, and all participants will be pushed to focus on their tasks with appropriate effort. Thus project status, challenges and progress are transparent to all research collaboration members at any time.

***Keywords*-Research Best Practices; Research Collaboration Management; Metrics; Lean Software Development**

I. Introduction

Collaborative research projects between software engineering researchers and practitioners provide opportunities for both sides: researchers influence their communities with new insights and practitioners integrate researchers' results into new business models, generating more business value.

Forced by their daily business needs, practitioners usually focus on time-to-market and design-to-budget strategies. This focus forces them to look for quick solutions that enable their business models with limited resources. As a result of those strategies, some solutions lead into a technological dead end (solution does not scale, consumes too many resources, or does not fulfill business case). In these situations, practitioners look for collaborations with research institutes, expecting solutions for their hard problems.

However, collaborations between researchers and practitioners carry the risks of organizational or cultural conflicts. Based on practitioners' mindset, they are afraid, that researchers focus on theoretical challenges which do not have any impact on the business value in the short or medium term. Researchers are afraid that practitioners focus on quick solutions too much, and that those solutions will not solve the challenging parts of their business case. Section II provides a detailed view on these risks and related work.

This paper provides our **Me**trics **Dri**ven Rese**a**rch Collaboora**tion** (MEDIATION) approach which tackles the risks of collaborative research projects. Our approach ensures continuously that all project participants have a common objective: the success of the project. The approach enables this clarification by making the project status, challenges, and progress transparent to all participants at anytime (c. f. Section III).

The approach has been evaluated on the German collaborative research project ETL Quadrat [1]. ETL Quadrat targets the development of an extended and semantic Extract, Transform, and Load (ETL) process which handles structured and unstructured, especially human-readable, data. The project involves two medium-sized companies and one research institute. In the beginning the mentioned problems along with misunderstandings arose and put the project success at risk. After the introduction of our approach, the research collaboration focused on common research goals which helps to reduce the risk of failure—Section IV provides detailed information and Section V concludes the paper.

II. Mutual Expectations of Research and Industry, and Their Potential Conflicts

Collaboration of practitioners and researchers generates a set of mutual expectations. This section provides an overview of expectations and potential conflicts from the practitioner's and researcher's view.

A. Practitioners' Expectations

Practitioners usually have to prioritize their effort on strategies like time-to-market or design-to-budget because their expenses, involved in their day-to-day business, forces them to be profitable with new products as soon as possible.

Scrum or Kanban support them to achieve shorter time-to-market slots [2]. Practitioners' mindset is minted by the Pareto principle: acquire 80% of potential costumers with 20% of the effort and avoid acquiring the remaining 20% of customers that requires 80% effort.

This attitude raises the following question: is it possible that practitioners keep being scalable or competitive by looking only for easy solutions? Often, practitioners notice that their software systems are going to reach an impasse regarding performance or quality, or even a difficult use case cannot be implemented. In those cases they start exploring the hard and challenging problems often supported by collaborations with researchers.

Given the practitioners' Pareto mindset it is clear what they expect from a research collaboration: a successful solution to their business problem which makes them more productive, cheaper, or acquires more customers. Additionally, they expect a constant progress which should be reached as quick as possible.

B. Researchers' Expectations

Researchers are always interested in real world question, but they primarily look for hard problems to provide contributions to their research field and to extend the state of technology.

This attitude raises following question: do researchers focus on hard challenges too much with the risk that such problems are not relevant to industry? Often, researchers implement complex technological applications which are based on their deep technological know-how. However, in some cases the solution needs to be evaluated on a business case supported by collaborations with practitioners.

Given the researchers mindset it is clear what they expect from a research collaboration: interesting and challenging problems which are hard to solve and which need deep investigations. They strive for new insights for themselves and for high quality contributions to their research community.

C. Potential Conflicts

The expectations from both sides can result in organizational conflict situations, blaming each other for not providing adequate solutions to the collaboration. We observed that such situations can lead to win-lose conflicts which have negative effects, for example deadlocks or delayed decisions. Pfeiffer and Staff [3] provides detailed information about those situations.

One of the main reason for this observation originates from a missing common understanding of the project goals. For example, most research collaboration have defined a common plan (e. g. it is a requirement for funding), but this early plan does not define the requirements explicitly. Based those inexplicit requirements each participant can interpret the collaboration goals independently and differently. Additionally, these inexplicitly defined requirements are not measurable. In consequence, the chasm increases between participants and researchers because of their different mind sets. To put it in exaggerated terms:

- Practitioners might think that researchers just pick interesting and hard challenges which do not solve the practitioners' business case.
- Researchers might think that practitioners apply only the Pareto principle which does not extend current state of technology.

D. Related Work

Fraser and Mancl [4] summarize how research collaborations can succeed and which strategies are useful to transfer innovations from collaborations into companies business models. For example, tracking objectives and agile development techniques are key factors of a successful research collaboration. They state that the *"most important first step is to create a structured way to share information"* [4] which is supported by our approach. Additionally, our approach supports *"improvements in internal development practices"*, *"tracking objectives"*, and *"determining the return on investment"*.

Our approach is based on metrics and it is important that those metrics match precise properties. Fowler [5] states that software development metrics are often used inappropriately and he provides guidelines for appropriate use: link metrics to goals, favor trends, short tracking periods, and change metrics when they stop driving change. Our approach considers those guidelines. Fowler [5] analyses the use of software development metrics, however those guidelines should also apply to metrics measuring business value.

Reinertsen [6] argues that the metric "cost-of-delay" should be the leading reference for lean product development. Leffingwell [7] also argues that the business value serves (incrementally measured) as the most important prioritization criteria. Cao and Ramesh [8] discuss, if focusing on business value is sufficient, because technical aspects are not considered at all. We extend the approaches of Leffingwell and Reinertsen by a small number of application-oriented metrics which serve as a middle layer between code base and the business-oriented view, mediating between development and management team. The definition of application-oriented metrics requires that the project team agrees to common goals. Additionally, metrics provide more transparency about the project status.

III. Metrics Driven Research Collaboration Approach To Keep The Common Objective in Focus

Our approach is called **Me**trics **Dri**ven Rese**a**rch Collaboration (MEDIATION). According to the name, the definition of metrics is the most important task in a research collaboration; the definition of metrics brings common goal definitions into focus. Each metric links back to the global project requirements and therefore all metrics are application-oriented. The research collaboration should elaborate a consensus on a small number of high-level metrics which put the most important requirements into focus. Fig. 1 illustrates the lifecycle of the MEDIATION approach.

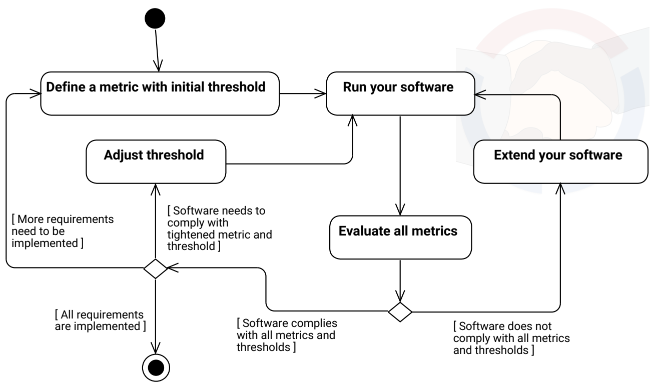

Fig. 1. Continuous Life-Cycle of Defining and Adjusting Metrics

At first, all participants should choose a requirement and define a corresponding metric altogether. This metric should have following properties. The metric should

- represent a common high-level business goal,
- be evaluated automatically,
- correspond with project status,
- have an initial threshold,
- and have a rich set of example input and output data.

For example, a sales company needs to extract product attributes (price, name, amount, etc.) from the web. The metric should compare the example output with actually extracted output data, and count the correctly extracted attributes. This metric links back to the requirement, that this company needs all product information as precise as possible. An initial threshold could be one correctly extracted attribute per web page.

When the participants defined this metric, they run their software and evaluate the metric. If software does not comply with the metric and initial threshold, the participants need to extend their software until it complies with the metric.

When the software complies with the metric and threshold, the corresponding business goal is achieved. New requirements and corresponding metrics will be added. The participants should check again if the software complies with all metrics and should extend the software if necessary. Parallel to defining new metrics, the participants can adjust thresholds of metrics to tighten the requirements. This process is repeated until all requirements have corresponding metrics and the software complies with all of them.

By following our MEDIATION approach researchers and practitioners will gain a rich set of benefits (see Section III-A). An appropriate tooling makes the status and progress of the research collaboration transparent to every participant (c. f. Section III-B). Additionally, this approach scales with multiple sub-projects (c. f. Section III-C).

A. *Benefits of Metrics Driven Research Collaboration*

Through our MEDIATION approach the project team focuses on the most important requirements and checks continuously if the software product complies with the corresponding metrics. This setup enables a set of benefits for the research collaboration.

1) Minimum Viable Product: Based on the defined metrics the research collaboration can choose to build the Minimum Viable Product (MVP) in the first iteration of MEDIATION. The MVP is a tactic which reduces the risk of wasted engineering effort by developing a software with a minimal feature set which is evaluated by visionary customers (c. f. Lenarduzzi and Taibi [9]). The research collaboration can operate with this tactic to evaluate the software during the whole project.

2) Iterative and Incremental Software Development: MEDIATION enables iterative and incremental software development: in repeated cycles (c. f. Fig. 1) the research collaboration implements solutions in small portions—small portions can be the definition of a metric or the adjustment of a threshold. Based on earlier cycles and portions, the collaboration partners learn during the usage and development of the software.

3) Transparent Progress: Through MEDIATION the project status is transparent to every project member at any time. For every metric it can be observed if the metric improves, stagnates, or degrades. If a metric…

- …improves, the software product is getting better in complying with the requirements. More compliance with metrics means better project status.
- …stagnates, the compliance with the requirements stays the same. For example, refactoring tasks can cause stagnation.
- …degrades, the compliance with the requirements is getting worse. A degrading compliance shows that provided solutions do not relate to business cases which limits the solution space for the research collaboration.

4) Team Control: Due to the fact that MEDIATION measures compliance between software and requirements, every collaboration member can orientate on this measurement. This orientation allows the development and management team to prioritize tasks better.

B. *Apply Tooling for Metrics Driven Research Collaboration*

In order to gain all the benefits of our MEDIATION approach it is important to apply an appropriate tooling. This tooling ensures that the current status of the research collaboration is transparent to all collaboration members. Transparency will be achieved by visualizing all metrics in a common place, requiring following steps and tools:

1) Build an Early Prototype: It is important that all metrics are executed on actual data. Therefore, it is necessary to build a running prototype after defining the first metric. This prototype can be based on Docker[1] images, because they provide advantages regarding reproducibility for both researchers and practitioners [10]. The first prototype version can provide random data: it is just used for defining all interfaces and it ensures that metrics will be evaluated.

[1] https://www.docker.com/

2) Build a Continuous Integration (CI) Pipeline: The prototype should be build by a CI pipeline on a regular basis (after each commit). Additionally, the source code should be tested by unit tests and quality analysis should be enabled as well. Therefore, we recommend use of Gitlab[2] which integrates CI easily into source code management.

3) Use an Acceptance Test Driven Development Framework: When the prototype is build by a CI pipeline, a Acceptance Test Driven Development (ATDD) framework should be integrated (e. g. jbehave[3]). With such a framework the project team can define every metric (the requirements) in a human-readable format which is mapped to the ATDD test code, executing and evaluating all metrics. The results of all metrics should be pushed to a time series database (e. g. InfluxDB[4]), in order to measure improvement, stagnation, or degradation.

4) Visualize Metrics on a Dashboard: Finally, all metrics will be visualized. In order to visualize metrics connect a dashboard software (e. g. Grafana[5]) to the time series database and configure the dashboard software to visualize the metrics. Additionally, the dashboard software should contain alerting mechanisms.

The dashboard is the reference for every collaboration member and it provides transparency of project progress and challenges at any time. This dashboard can be compared to Scrum project boards which provide progress and challenges to all participants in a similar way.

Fig. 2 illustrates an example setup how our approach evaluates the metrics automatically. When the CI pipeline executes the ATDD scenarios, the framework takes the example input data and runs the software, generating actual output data (c. f. $Input$ and $Output$ in Fig. 2). After that, a metric $\Delta = ||Output - Output_{gold}||$ describes how well the software produces the actual output data compared to the expected output data (c. f. $Output_{gold}$ which represents the optimal output of a test case). Δ describes the compliance between $Output$ and the defined requirements.

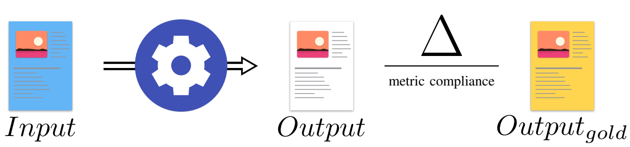

Fig. 2. Automatic Metrics Evaluation For Compliance Determination

C. Scale Metrics Driven Research Collaboration by Multiple Sub-Projects

Our MEDIATION approach can be applied to sub-projects as well. The application to sub-projects provides additional benefits, providing more insight and transparency of the project status:

- The influence among all sub-projects can be measured: If a sub-project relies on the intermediate results of another sub-project, it is possible to determine if the intermediate results are sufficient to fulfill the global project requirements.
- The participants of a sub-project can work independently from other sub-projects: The participants can focus on their requirements and metrics to optimize them. When each sub-project complies with its own requirements, the final software product complies with the global requirements.
- The utilization of resources can be improved: If a sub-project has difficulties in fulfilling their requirements, the management team of a research collaboration can decide to take resources from other sub-projects and put the resources into this specific sub-project.

The following example illustrates how a research collaboration can achieve those benefits.

Fig. 3 shows the automatic evaluation of metrics for a software product which is based on pipes and filters architecture. If a software product can be decomposed into multiple parts, the following set of ATDD test scenarios can provide information of mutual influence. The whole software product consists of three decomposed steps (*Sub-Project A*, *Sub-Project B*, and *Sub-Project C*). Each sub-project has reference input and output data: *Sub-Project A* has input data I_1 and output data I_2; *Sub-Project B* has input data I_2 and output data I_3; and *Sub-Project C* has input data I_3 and output data O_{gold}.

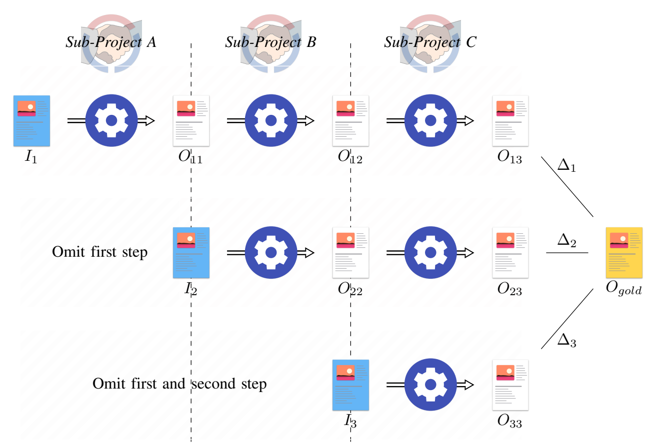

Fig. 3. Automatic Metrics Evaluation For Compliance Determination with Multiple Sub-Projects with Pipes and Filters Architecture

This setup allows to measure the progress and compliance of each sub-project. Furthermore, it is possible to measure the influence among the project as well. Therefore, a set of ATDD test scenarios will execute the software differently.

- The first scenario executes all decomposed steps one after another. The scenario passed I_1 to the first step and it passes the result of the first step to the second step and so forth. At the end the scenario can evaluate Δ_1 which shows the overall compliance of the whole software

[2]https://about.gitlab.com/
[3]http://jbehave.org/
[4]https://github.com/influxdata/influxdb
[5]http://grafana.org/

product.
- The second scenario executes the decomposed steps and omits the first step. Δ_2 describes the compliance with the requirements. The difference between Δ_1 and Δ_2 describes how big is the error propagation of the first step.
- The third scenario omits the first and second step in the execution of the software product. $|\Delta_3 - \Delta_2|$ (not depicted) describes how big is the error propagation of the first and second step.
- Furthermore, those scenarios allow to measure the compliance at any point in the software pipeline. For example, $\Delta_{22} = ||I_3 - O_{22}||$ measures the compliance of the second step without error propagation of the first step.

This setup of ATDD scenarios allows one to discover how sub-projects influence each other. When this influence is known to all participants of the research collaboration, the research collaboration can benefit from the aforementioned advantages and it is very likely that the collaboration will develop a successful software product.

IV. Apply Metrics Driven Research Collaboration to ETL Quadrat

We developed our MEDIATION approach during the research project ETL Quadrat [1]. ETL Quadrat is a research collaboration with multiple partners: two medium-sized companies and one research institute. The research collaboration has following properties:

- ETL Quadrat extends the established ETL process of the medium-sized companies with an Ontology-based Information Extraction (OBIE) system, illustrated in Fig. 4. The companies extract product information from structured data sources and additionally they need to extract information from human-readable documents. For the latter they need a system which can handle natural language text which is performed by the OBIE system.
- The architecture can be decomposed into multiple independent steps and follows the pipes and filters architecture. Each research partner develops a different step in that architecture:
  - Homogenization: different document types (e. g. PDF, HTML, Word, etc.) need to be converted into a common format: OpenDocument Text (ODT). This reduces the complexity of following steps. Additionally, each step works on ODT files which is the exchange format among all steps.
  - Natural Language Processing (NLP) of OBIE: To Handle natural language text it is necessary to process the text with multiple NLP tools [11, Chapters 10, 11, and 12]. Those tools derive a structure from the unstructured natural language text which is required by the semantic interpretation.
  - Semantic interpretation of OBIE: Based on the derived structure the semantic interpretation transforms the structure into meaningful data [11, Chapter 25]. Which data needs to be extracted is defined in the global requirements of ETL Quadrat.

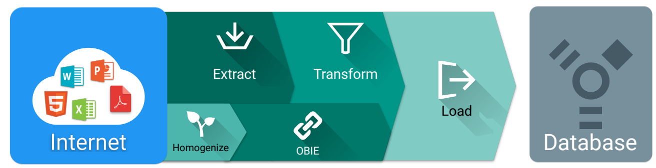

Fig. 4. Existing ETL Process, Enhanced by Semantic Interpretation (c. f. Homogenization and Ontology-based Information Extraction)

A. Metrics for Global and Sub-Project Requirements

According to our MEDIATION approach the first step is to define the metrics which reflect the global requirements of ETL Quadrat. Following examples illustrate some metrics based on our global requirements, true positives, false positives and false negatives of:

- required attributes for a set of example documents. Product name, price, components, amount, etc. are examples of required attributes.
- relations between the required attributes. For example, each component of a product is sold in a certain amount, therefore a extracted component refers to an amount.

Based on such metrics we can measure F-Score, and we can visualize those metrics on our dashboard which will keep all research participants informed about progress. Additionally, we define metrics for our sub-projects. For each sub-project we define corresponding input and output ODT files as reference. Following list illustrates some metrics based on the ODT files.

- Homogenization: One of the metrics for the homogenization sub-project measures the paragraph-based edit distance. This metric ensures that all characters in all paragraphs are converted correctly. For example, PDF files often contain special characters (mathematical symbols, etc.), which should be converted into the right UTF-8 encoding within the ODT files. Ideally, each paragraph has an edit distance of zero.
- NLP tools of OBIE: The reference ODT archives contain additional XML files which contain the expected structure of natural language text. Based on this data standard metrics (precision and recall, or accuracy) can be applied. The NLP tools of OBIE require domain specific training which is aided by another CI approach [12], [13].

B. Tools Improving Transparency of Challenges and Progress

Based on the mentioned metrics we establish a tool set. This tool set provides the transparency regarding progress and challenges.

1) Acceptance Test Driven Development Framework which Evaluates Metrics: All metrics are formulated in Gherkin language [14] which provides a readable format for all participants (not just for developers). Listing 1 provides an excerpt which illustrates a metric for the homogenization step.

Listing 1. Paragraph-Based Edit-Distance Metric Formulated in Gherkin

```
Feature: Convert documents from the WWW into ODT documents

Narrative:
In order to extract product information from web documents correctly
As a information extraction development team
I want to convert PDF documents into ODT documents as precisely as possible.

Scenario: The homogenization will be executed with multiple PDF documents

Given a <pdfDocument>
When the document has been converted into ODT
Then the average paragraph-based edit distance should be less than <avg> compared to expected <odtDocument>
And the maximum paragraph-based edit distance should be less than <max> compared to expected <odtDocument>

Examples:
| pdfDocument | avg | max | odtDocument |
| doc1.pdf    | 4   | 10  | gold1.odt   |
| doc2.pdf    | 1   | 5   | gold2.odt   |
```

If the requirements are formulated in Gherkin, they can be glued to the test code which evaluates and stores the metrics. In our setup we use jbehave as ATDD framework, InfluxDB, and Spring Data InfluxDB[6]. The test code which corresponds to Listing 1 is illustrated in Listing 2.

Listing 2. Java Test Code which Evaluates and Stores Metrics

```
@Autowired
private InfluxDBTemplate<Point> influxDBTemplate;

// ...

@Then("the average paragraph-based edit distance should be less than <avg> compared to expected <odtDocument>")
public void avgEditDistanceShouldBeLessThan(@Named("avg") int avg, @Named("odtDocument") String odtDocument) {

  // Determine avg edit distance
  int actualAvg = determineAvgEditDistance(odtDocument);

  // Store metric in InfluxDB
  Point p = Point.measurement("homogenization")
    .time(System.currentTimeMillis(), TimeUnit.MILLISECONDS)
    .tag("document", odtDocument)
    .addField("avg-paragraph-based-edit-distance", actualAvg)
    .build();
  influxDBTemplate.write(p);

  // Make sure that metric complies with threshold
  assertThat(actualAvg, is(lessThan(avg)));
}
```

[6]https://github.com/miwurster/spring-data-influxdb

C. Evaluation

During the research collaboration of ETL Quadrat we developed our MEDIATION approach to achieve a better common understanding of the research goals. During this time practitioners and researchers acted as developers. Furthermore, researchers acted as consultants to the project tasks which have been developed by practitioners. The following situation predominated (without MEDIATION):

1) Defuse Common Goal: Every project member had a very different opinion which attributes need to be extracted effectively.

2) No Transparent Progress: Without the example data, it was unclear how good the software performs.

3) Lack of Team Control: Every team member had difficulties to choose which tasks need to be implemented next.

Based on those problems we decided to build a dashboard which helps to guide development and eliminates mentioned problems. This dashboard is illustrated in Fig. 5. The dashboard is implemented through Grafana and the structure corresponds with Fig. 3.

- The first line shows the execution of the whole software pipeline. The output quality regarding attribute extraction

Fig. 5. Metrics Driven Research Collaboration Dashboard of ETL Quadrat based on Grafana and InfluxDB

is measured by a F-Score of 0.48.

- The second line shows the execution of the software pipeline without the homogenization step. The output quality is measured by a F-Score of 0.6. The error propagation through homogenization is 0.12.
- The last line shows the execution of semantic interpretation. With ideal input data for this step the F-Score is 0.8.
- Furthermore, the dashboard shows the error propagation between homogenization and NLP tools as well.

First of all, after the definition of metrics with corresponding threshold, every collaboration member knows which attributes and relations need to be extracted from the example documents. Each metric links back to a requirement explicitly (in our case formulated in Gherkin) which makes the business case transparent to every project member. Without those definitions, participants had very different opinions which attributes have to be extracted.

This dashboard provides a lot of insight into the state of the research collaboration. The development team prioritizes the tasks to optimize the overall metric compliance. In each development iteration (ETL Quadrat has a prototype demonstration every two month), the research collaboration gains insight with every prototype version, how to optimize the metric compliance. Each decision is based on actual data and decisions are not based on gut feelings.

After applying our MEDIATION approach the development and progress has become more effectively. Resources could be assigned to the problems which cause stagnation in metric compliance. Every participant was convinced that the developed software product improves the business value. Additionally, researchers focused on relevant challenges, solving the business case. The focus on business values through metrics reduced the risk of failure essentially.

V. Conclusion & Lessons Learned

In this paper we provide our **Me**trics **Dri**ven Rese**a**rch Collabora**tion** (MEDIATION) approach which treats metrics as a first-class citizen in research collaboration between practitioners and researchers. The requirements of collaborations constitute patterns for metrics, and metrics constitute common research goals. With this approach the status of research collaborations is made transparent to every participant, and participants are focused to solve problems with appropriate effort. Improving, stagnating, or decreasing of metrics are indicators for the success of research collaborations:

- Improving metrics are a strong indicator that the software complies with the requirements. The collaboration will succeed.
- Decreasing and stagnating metrics are a strong indicator that the software does not comply with the requirements. The collaboration might fail. However, management and development teams will be informed immediately through a dashboard, and they can initiate the necessary steps to succeed in fulfilling the requirements.

This approach requires a lot of ramp-up tasks: defining metrics based on requirements, setting up a CI pipeline with ATDD framework, setting up a time-series database and a dashboard. However, those ramp-up tasks are necessary to support research collaboration to succeed in the development. Additionally, there are many tools which make such a setup easy to install and reproducible [10]. In our case, all tools and metrics can be reused in the follow up project of ETL Quadrat, which shows that the MEDIATION approach is reusable.

References

[1] A. Kessell. (2012). ETL QUADRAT, Entwicklung einer Plattform-Technologie zur Produktdatenaufbereitung, [Online]. Available: http://www.pt-it.pt-dlr.de/_media/Infoblatt_ETL_QUADRAT.pdf (visited on 08/12/2016).

[2] K. Schwaber and J. Sutherland, *Software in 30 Days, How Agile Managers Beat the Odds, Delight Their Customers, And Leave Competitors In the Dust*. Wiley, Feb. 2012, ISBN: 978-1-118-20666-9.

[3] Pfeiffer and C. Staff, *The 1973 Annual Handbook for Group Facilitators*. John Wiley & Sons, Incorporated, 1973, ISBN: 978-0-7879-0380-0.

[4] S. Fraser and D. Mancl, "Strategies for Building Successful Company-university Research Collaborations," in *PROCEEDINGS OF THE 3RD INTERNATIONAL WORKSHOP ON SOFTWARE ENGINEERING RESEARCH AND INDUSTRIAL PRACTICE*, ser. SER&IP '16, Austin, Texas: ACM, 2016, pp. 10–15, ISBN: 978-1-4503-4170-7. DOI: 10.1145/2897022.2897025.

[5] M. Fowler. (Feb. 2013). An Appropriate Use of Metrics, [Online]. Available: https://martinfowler.com/articles/useOfMetrics.html (visited on 01/20/2017).

[6] D. G. Reinertsen, *The Principles of Product Development Flow, Second Generation Lean Product Development*. Celeritas Publishing, 2009, ISBN: 978-1-935401-00-1.

[7] D. Leffingwell, *Agile Software Requirements*. Addison-Wesley Professional, 2011, ISBN: 978-0-321-63584-6.

[8] L. Cao and B. Ramesh, "Agile requirements engineering practices: An empirical study," *IEEE Software*, vol. 25, no. 1, pp. 60–67, Jan. 2008, ISSN: 0740-7459. DOI: 10.1109/MS.2008.1.

[9] V. Lenarduzzi and D. Taibi, "MVP Explained: A Systematic Mapping Study on the Definitions of Minimal Viable Product," in *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Aug. 2016, pp. 112–119. DOI: 10.1109/SEAA.2016.56.

[10] J. Cito and H. C. Gall, "Using Docker Containers to Improve Reproducibility in Software Engineering Research," in *Proceedings of the 38th International Conference on Software Engineering Companion*, ser. ICSE '16, Austin, Texas: ACM, 2016, pp. 906–907, ISBN: 978-1-4503-4205-6. DOI: 10.1145/2889160.2891057.

[11] R. Mitkov, *The Oxford Handbook of Computational Linguistics*. Oxford University Press, 2003, ISBN: 978-0-19-927634-9.

[12] M. Schreiber, K. Barkschat, and B. Kraft, "Using Continuous Integration to Organize and Monitor the Annotation Process of Domain Specific Corpora," in *5th International Conference on Information and Communication Systems (ICICS)*, Irbid, Jordan: IEEE, Apr. 2014, pp. 1–6. DOI: 10.1109/IACS.2014.6841958.

[13] M. Schreiber, B. Kraft, and A. Zündorf, "Cost-efficient Quality Assurance of Natural Language Processing Tools Through Continuous Monitoring with Continuous Integration," in *PROCEEDINGS OF THE 3RD INTERNATIONAL WORKSHOP

ON SOFTWARE ENGINEERING RESEARCH AND INDUSTRIAL PRACTICE, ser. SER&IP '16, Austin, Texas: ACM, 2016, pp. 10–15, ISBN: 978-1-4503-4170-7. DOI: 10.1145/2897022.2897029.

[14] S. Rose, M. Wynne, and A. Hellesoy, *The Cucumber for Java Book, Behaviour-Driven Development for Testers and Developers*. 2015, ISBN: 978-1-941222-29-4.