TechSuRe
A method for assessing technology sustainability in long lived software intensive systems

Anton Jansen  
ABB Corporate Research,  
Industrial Software Systems  
Västerås, Sweden  
Anton.Jansen@se.abb.com

Anders Wall  
ABB Corporate Research,  
Industrial Software Systems  
Västerås, Sweden  
Anders.Wall@se.abb.com

Roland Weiss  
ABB Corporate Research,  
Industrial Software Systems  
Ladenburg, Germany  
Roland.Weiss@de.abb.com

Over the lifetime of a long lived software intensive system the (software) technologies used in the system will change. Changing technologies in such systems is typically extremely costly and painful for the organizations that have to perform these changes. Hence, wrong technology choices can jeopardize the economic viability of such systems in the long run. This paper presents TechSuRe, a method for assessing technology sustainability in long lived software intensive systems, together with experiences from case studies. The method makes sustainability issues an explicit part of a technology assessment and offers guidance on how to gauge the associated sustainability. Together, this supports making appropriate technology choices for long lived software intensive systems.

Sustainability; Economics; Risks; Technology

I. INTRODUCTION

Lifetimes up to 30 years are not uncommon for long lived software intensive systems, which typically out-live the parts from which they are built. To warrant such long lifetimes, these systems need to be sustainable. In this paper, we focus on the problem of economic sustainability, i.e. how can a system remain economically viable over its entire lifetime? In particular, this poses a problem for software intensive systems, as technology changes will happen frequently over the lifetime of the system, whereas from a risk and cost management perspective we want to minimize these changes. Hence, choosing the right set of technologies for a long lived system is crucial to let the system remain sustainable.

The contribution of this paper is TechSuRe, a method for supporting the reasoning about sustainability and its associated risk when introducing a new (software) technology into a software intensive system. The TechSuRe method provides a formalized set of steps to take, and tools to use to assess sustainability (risks). The method examines sustainability from three distinct perspectives: time, risk, cost and benefit. TechSuRe focuses primarily on the former two, although the latter is still in scope of the method.

The result from executing TechSuRe is an assessment report, which includes an indication on the technology expected lifetime and sustainability in a product. It is important to acknowledge that an exact quantitative results of an assessment is not the main target. Rather, the main purpose of the method is to ensure that sustainability aspects are raised and discussed in a technology assessment.

The remainder of this paper is organized as follows. Firstly, Section II presents related work with respect to technology assessment and sustainability. Section III introduces the TechSuRe method and the reasoning framework that forms its backbone. Thirdly, Section IV presents a questionnaire about the indicators used in the TechSuRe method, which is followed by three case studies that are described in Section V. The paper concludes with future work and a conclusion in Section VI.

II. RELATED WORK

In [8] measuring software sustainability and sustainment is discussed. They claim that software sustainability depends upon more than just the source code (technical aspects such as software architecture), it includes the organization sustaining the software, customers, and the operational domain in which the software operates.

Moreover, they claim that decisions about replacing technologies are subjective and should be based on a risk/benefit trade-off. Lack of experienced staff in a technology should not prevent it from being selected. This simply becomes an additional inhibitor to achieving the ideal solution that must be overcome by training or hiring, which is acknowledged by competence indicators in TechSuRe. Business models/rules are acknowledged as a factor influencing sustainability, which is also part of the estimation of technology sustainability risk in TechSuRe.

A similar reasoning is used for the domain of supply chain management systems in [6]. Here, the authors claim that there is a direct relationship between technology choices and the sustainability of a system. In particular, they stress the importance of road mapping as a tool to identify sustainability issues. This aspect of sustainability is reflected in TechSuRe, as the method makes explicit the expected evolution of a technology by the use of scenarios.

A framework for estimating software technologies is presented in [1]. This framework is mainly focused on comparing technologies on a feature basis, i.e. functionality provided by the technologies and its distinctive features in
relation to other technologies. Moreover, the framework embraces the inherent complexity, ambiguity, and dynamism of the technology marketplace and that features offered by competing technologies change over time. The comparable technology identification used for establishing lifetime of technologies in TechSuRe is influenced by the ideas of domain genealogy and comparative used in [1]. TechSuRe also acknowledges that features are changing over time and that technologies have a lifecycle themselves by estimating a technology evolution indicator, by looking at risks associated to a technology’s market and by estimating the technology’s fitness for use in a specific product over time.

There exist more formal methods for assessing lifetime of products in general. However, products in general are often considered physical products that wear out physically over time which is not the case for software. In [10], the authors divide lifetime of a product into two different classes: the physical lifetime and the value lifetime. Value lifetime is influenced by a product’s ability to fulfill a customer’s needs regarding performance, functionality, and appearance. The value lifetime is applicable to software in our framework.

One approach to estimate this value is the Net Present Value (NPV) approach. It comes from the world of corporate financing and is an accounting method for valuating the capital and budgeting of projects. With respect to software, the method can be used to predict the economic value of different alternatives, as to provide a basis for strategic decision making. A nice introduction from a software perspective to the method is given in [3].

Once a technology has been chosen, additional measures have to be taken to ensure sustainable use of the technology. Software evolution [6] addresses this need. In [10] a literature survey classifies software evolution by means for identification and analysis as well as solutions to evolution challenges.

Overall, our approach bridges the areas of technology evaluation, value assessment, and software evolution with a tailored solution for selecting technologies based on sustainability indicators.

III. TECHSURE

A. Introduction

The aim of the TechSuRe method is to give a technology assessor a way to systematically assess the long term economic impact and associated risks that introducing a software technology pose for a particular long lived software intensive system. When a choice has to be made among several technology options, the method provides the technology assessor with a much broader scope for supporting a technology choice than a traditional price and feature comparison would give. To ensure the adaptation of the method by technology assessors the method should be easy to use and performing an assessment using the method should not take more than a one day including preparations.

The TechSuRe method uses a bottom-up approach to make its assessment. The method starts with the assessor quantifying different indicators on a three-level discrete scale (e.g. high, medium, low). Next, a reasoning framework is provided that, given the quantification of the individual indicators, guides the technology assessor to reach conclusions.

The reasoning framework is represented as a set of tables where the left column holds the results of the reasoning given the indicators. We have not ranked the importance of each indicator, this is implicitly handled in the resulting estimate.

To exemplify how our reasoning framework is designed see the reasoning framework presented in TABLE I. The example illustrates the choice of protection to use in different weather conditions. The two indicators in this small example that provides the basis for this decision are precipitation and wind condition. For instance, if it is raining and the wind speed is moderate, an umbrella is probably a good choice. However, if precipitation is high, i.e. it is raining hard, and the wind speed is rather high it is probably better to put on a raincoat.

By estimating and quantifying the indicators in the table, the result can be tracked in the result column (weather protection). In this way, we embed the reasoning and intelligence about what certain combinations of indicators means.

<table>
<thead>
<tr>
<th>Weather protection</th>
<th>Precipitation</th>
<th>Wind speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Raincoat</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Umbrella</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Raincoat</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Umbrella</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Umbrella</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>Raincoat</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

The result from applying our reasoning framework is a statement about the sustainability and associated sustainability risk for using a particular software technology in a specific software intensive system. Note, that the framework still depends on having a non-biased and knowledgeable technology assessor, as his or her expert judgment is required to assess the value of each.

B. Global picture

Figure 1 presents a high-level overview of the structure of our reasoning framework illustrating the scope of the reasoning framework including all high-level indicators and their relation to sustainability (risk). The method for estimating cost and benefit is, although important for sustainability, left out of the scope. The method leaves this to other existing approaches, e.g. Net-Present Value [3], Return-on-investment, etc.
As depicted in Figure 1, *Sustainability risk* is estimated based on nine high-level indicators: *Lifetime in production*, *Lifetime*, *Competence risk*, *Technology evolution risk*, *Risk of changing business model*, *Market risk*, *Lifetime certainty*, *Complexity risk*, and *Technology evolution fitness*.

*Lifetime in production* is the expectations we have concerning lifetime of the software technology. This is very product specific, i.e., we may expect shorter lifetimes of technologies used in cell phones than we do in control systems for nuclear power plants. Moreover, expectations can vary depending on in which part of the system the technologies are used. As an example, we have found that technology changes are much more frequent and accepted in the user-interface of a system, than it is in its core business logic.

*Lifetime* is an estimation of the expected lifetime for a specific software technology. This assessment is done by comparing similar existing or pre-existing technologies and estimating how the lifetime of the technology under evaluation measures up compared to these other technologies.

*Competence risk* is concerned with how difficult it will be to acquire the competence needed to manage the technology, e.g. through training, the possibility to hire a person with necessary skills, etc.

*Technology evolution risk* is concerned with the effort of following the lifecycle of the technology, i.e. upgrade or back-wards compatibility, and also the effort of exchanging the technology for another in the future.

*Risk of changing business model* captures the problem that the technology vendor may change the business model for the technology in the future. Examples are changes in licensing strategies, going from an open to a proprietary approach, etc. If the business model of a particular technology does not fit the product and the way a product company does business, it is as discriminating as having a technology that does not fit the product functional-wise.

*Market risk* in TechSuRe is concerned with whether the technology complies with a standard, and the quality of that standard. Moreover, market share of the technology and our possibilities to influence the technology is important. If competitors are using the same technology in their products it is an indication that the risk is rather low since if the technology gets obsolete, we are not in a worse position than our competitors. However, using the same technology as our competitors may also indicate that the possibilities for market differentiating features for our-, and our competitors products are limited.

*Lifetime certainty* is an indication on how sure we are about the estimated lifetime. The precision with which we can estimate the lifetime, i.e. lifetime certainty, is dependent on the maturity of the technology and the maturity of the problem domain it serves. The market adoption is another indicator contributing to the certainty of our lifetime estimation. Typically, if the market adoption is very low, there may not be a large interest from the vendor to adhere to the expected lifetime plan.

It is very difficult to objectively express the complexity of a piece of software. In TechSure we use a subjective estimation based on experiences with the technology and possibly earlier experiences with similar technologies to assess *Complexity risk*.

*Technology evolution fitness* is about estimating how well a technology suits a product’s future requirements. TechSuRe uses a scenario-based approach to estimate technology evolution fitness by deriving scenarios that reflects possible future directions for a product for which the technology under evaluation influences.
Figure 2. The TechSuRe indicators and their relationships
The high-level indicators depicted in Figure 1 are, in turn, estimated by using more fine-grained indicators, which are presented in Figure 2. The high-level indicators are presented with a gray background, intermediate indicators with a white background, and input indicators with a black background. Similar to the high-level indicators, the intermediate indicators are derived from other indicators (i.e. from other intermediate indicators or input indicators). The input indicators are the indicators that a technology assessor has to assess. Based on this assessment, the reasoning framework will guide the estimation of intermediate and high-level indicators.

C. An example of an indicator

In this subsection, we will explore one of the indicators of the TechSuRe: lifetime and partially show how this indicator comes together in the sustainability risk estimation. Figure 3 presents a breakdown of how the lifetime indicator is built up. The technology assessor needs to assess five indicators, i.e. release frequency, business model, maturity, supplier's dependency on technology, and market adaptation. Next, using the aforementioned reasoning tables, the technology assessor determines the technology viability and suppliers technology commitment, which in turn are used to assess the expected lifetime of the technology. In the remainder of the example, the focus will be on one branch of the lifetime indicator, i.e. we will focus on how the supplier technology commitment is assessed and how this contributes to a judgment about the sustainability risk.

Figure 3. The breakdown of the lifetime indicator

The first indicator to assess for the supplier technology commitment is the supplier's dependency on technology. A supplier may have several different product-lines in its product portfolio, each with different product strategies and roadmaps. Typically, the major revenues for a software technology supplier are in a single, or a small set, of products. Technologies, which are marginal to the supplier’s revenues, will probably stand less chance to survive in the long run than the bigger product-lines. The supplier’s dependency on the technology is the ratio of revenue of the supplier that comes from the technology. An example of a low supplier’s dependency on technology was Microsoft’s Visual Basic, whereas an example of a high dependency was VxWorks for Windriver. This information can typically be found in financial reports if the supplier is on the stock market.

The second indicator is the market adoption, which is the market share the technology under evaluation has in its domain as compared to competing technologies. A rough rule of thumb for assessing market adoption is the following:

- Low if market share is less than 10%
- Medium if market share is between 10% and 30%
- High if market share is more than 30%.

Next, the supplier technology commitment can be assessed, which is the probability that the supplier will keep on developing and supporting the technology. To do so, the technology assessor uses the values of the market adoption and supplier's dependency on the technology indicators together with the TechSuRe reasoning framework to derive a value for the supplier technology commitment. TABLE II presents the reasoning framework for this indicator.

In a similar fashion, the technology viability is assessed. Next, we combine the technology viability with the supplier technology commitment to estimate the expected lifetime of a technology (see TABLE III.).

TABLE II. SUPPLIER’S TECHNOLOGY COMMITMENT

<table>
<thead>
<tr>
<th>Supplier Technology commitment</th>
<th>Market adoption</th>
<th>Supplier’s dependency on the technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Medium</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Medium</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Low</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>Medium</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>Medium</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

TABLE III. LIFETIME

<table>
<thead>
<tr>
<th>Lifetime</th>
<th>Technology viability</th>
<th>Supplier Technology commitment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Low</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Medium</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

To assess the sustainability risk, the lifetime is combined with the lifetime in production to assess the technology lifetime fitness (see Figure 1). This and other high-level indicators influence the sustainability risk with some of
them being more dominant than others. This is acknowledged in TechSuRe by assigning weights to these individual high level indicators. Table IV presents not only these weights, but also presents how the qualitative estimates of each indicator are translated into quantitative numbers. The reason for this is that the sustainability risk is determined by using the weighted sum of its individual indicators.

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Estimation</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lifetime certainty</td>
<td>High = 1, Medium = 2, Low = 3</td>
<td>3</td>
</tr>
<tr>
<td>Technology evolution fitness</td>
<td>High = 1, Medium = 2, Low = 3</td>
<td>3</td>
</tr>
<tr>
<td>Risk of technology changing business model</td>
<td>High = 3, Medium = 2, Low = 1</td>
<td>1</td>
</tr>
<tr>
<td>Technology evolution risk</td>
<td>High = 3, Medium = 2, Low = 1</td>
<td>2</td>
</tr>
<tr>
<td>Competence risk</td>
<td>High = 3, Medium = 2, Low = 1</td>
<td>2</td>
</tr>
<tr>
<td>Complexity risk of the technology</td>
<td>High = 3, Medium = 2, Low = 1</td>
<td>2</td>
</tr>
<tr>
<td>Market risk</td>
<td>High = 3, Medium = 2, Low = 1</td>
<td>2</td>
</tr>
<tr>
<td>Technology lifetime fitness</td>
<td>Poor = 3, Reasonable = 2, Good = 1</td>
<td>3</td>
</tr>
</tbody>
</table>

After transforming the indicators into quantitative numbers the sustainability risk value is calculated using the formula:

\[
\text{Sustainability Risk Value} = \sum (\text{Estimation(Indicator)} \times \text{Weight(Indicator)})
\]

Given the sustainability risk value, we classify the sustainability risk as follows:

- **High** if >38
- **Medium** if 27 <= Sustainability risk value <= 38
- **Low** if < 27

Note, that the precise numbers are not the main contribution of the method, as the cut-off numbers may vary per company and/or domain. Instead, it is the process of analyzing a software technology from a sustainability point of view that is the real strength of the method.

IV. QUESTIONNAIRE

Validating the TechSuRe method with a high degree of confidence is due to its breath in scope and the long term in which sustainability aspects surface quite difficult. In this paper, we report on two different approaches to partially validate the method. The first approach is a survey, which focuses on the credibility of the indicators used in TechSuRe and is presented in this section. The second approach, presented in Section 5, focuses on the suitability of the method, as it describes three case studies in which the TechSuRe method is applied.

A. Introduction

The survey was developed with the aim to answer two particular questions:

- Do we cover all the important indicators in TechSuRe?
- What is the importance of these indicators?

The first question focuses on how complete our method is, i.e. do we have any obvious gaps? The second question focuses on the correctness of our understanding, i.e. do we indeed focus on the most important/influential indicators? To investigate these questions, we created an anonymous exploratory web survey.

Besides three subject profiling questions, the main part of the questionnaire covered the top level indicators used in the TechSuRe method (i.e. lifetime, lifetime certainty, market risk, changing business model, technology evolution fitness, technology evolution risk, and competence risk). For each of these indicators, we asked the respondents a question of the form:

- **How much does X influence sustainability risk**
  - (1 = no influence, 10 = dominant influence)?

    where X is one of the top level indicators. The respondents can answer the question on a scale of 1 to 10, with 1 indicating that the indicator has no influence and a 10 indicating that the indicator is dominant. Besides this question, we provided a definition of each indicator, so our respondents knew the meaning of each indicator. Next, we focused on the first research question, by asking:

- **The previous set of questions asked about the influence indicators have on sustainability risk. Are these the most significant indicators with respect to sustainability risk? (Yes/No)**

    If a respondent answered “No” then an additional textbox was shown in which the subject should describe the missing factors in his/her opinion and rate their importance.

Over one hundred persons were approached in an e-mail to participate in the survey and asked to send on the e-mail to anybody they knew that might have an interest in the topic. The initial population of one hundred people includes architects from our company, and representatives from Swedish universities, institutes, and companies related to software intensive systems, and academic software architecture researchers. For a more complete description of the survey, we refer to [4].

B. Results & Analysis

In total, 32 people filled in the survey. 27 work in industry, 10 in academia, and 3 in a research institute (8 people work in 2 contexts instead of 1). Figure 4 presents an analysis of the results to the first set of questions about the influence of indicators. For each indicator, the average of the responses is presented, whereas the vertical error bars indicate the standard deviation of the underlying distribution.
With respect to Figure 4, we can make the following observations:

- The perceived importance of each indicator differs considerably among the subjects (as can be seen from the standard deviations)
- The risk of changing business model is considered to be less important than all the other indicators apart from the competence risk.
- Apart from the risk of changing business model, all the mentioned indicators are perceived to be more influential than average (all the averages are above 5.5, the expected average on a 1-10 scale).

On the question about completeness, 4 out of the 32 respondents indicated that the presented set of indicators was not complete and suggested to include the following indicators: engineering practice (i.e. who is in control of development), functional growth, data risks, and social / environmental / governmental sustainability goals. The engineering practice is indeed currently not covered in TechSuRe, as it assumes to be deployed in a technology company. The functional growth and data risks are covered in technology evolution fitness and risks indicators respectively. The last factor of social / environmental / and governmental sustainability goals are relevant, but out of scope for TechSuRe, as it focuses on economic sustainability.

A Threat to internal validity of this survey can be found in having ambiguous questions and/or definitions in the survey. We deal with this threat in two distinct ways. Firstly, we test run the survey in three iterations on three different test subjects to identify such problems beforehand. Secondly, we collect comments from our subjects with respect to the survey to identify such problems afterwards. With respect to external validity the threats to validity are twofold. Firstly, there is a bias in the in the subject population towards Sweden due to the way the initial seeding has been performed. Building up a broader, more international oriented, initial subject population might be worthwhile to counter cultural and national influences. Secondly, certain factors suggested by the subjects were not included in the survey, e.g. the engineering practice and social goals. The absence of these factors might have changed the perceived influence of the presented factor.

Coming back to the first research question the survey started with, we can conclude the following: TechSuRe seems to have covered most of the important indicators. In the feedback on completeness, the following two indicators were suggested: the engineering practice, i.e. the power balance between marketing/sales versus the technical side of an organization, and the data risks, i.e. the risk of changing data as the business environment changes. In future work, we will explore these factors and their relevance in more detail. The second question, i.e. the influence these factors have, is less clear to answer. The answers of our subjects differ quite substantially. The statistically relevant conclusions are twofold. Firstly, the risk of changing business model is considered to be less influential than all the other indicators apart from the competence risk. Secondly, apart from the risk of changing business model all the mentioned indicators are perceived to be more influential than average. What we found most surprising was the low influence of the risk of changing business model. It seems that people judge the risk of this happening rather low and consequently judge its influence as limited. For a more detailed description of the survey, the source data, and a statistical analysis, we refer to [4].

V. CASE STUDIES

We have applied the TechSuRe method described in Section III in three different technology case studies performed in ABB. In this section, we present these three case studies together with their lessons learned.

A. User Interface Technology

The first use of TechSuRe took place in the context of a technology pre-study for choosing an UI technology to be used in a software product family that would encompass several products that currently each have their own particular UI technology. Hence, instead of one product context, several contexts needed to be considered when applying the TechSuRe method. The method was applied at the end of the pre-study, when a choice had to be made between two remaining technologies. Consequently, the method was applied twice, once for each of the two technologies.

A week before the first TechSuRe assessment a short presentation (30 minutes) was given to the assessment team (3 people) to introduce the method. After the presentation, the team was e-mailed the first version of the TechSuRe method description. The purpose of this was twofold: to proof read the method description and for the team to learn the method. A week later the assessment took place. The first day, we provided guidance on using the method and the first technology was assessed. On the second day, without our guidance, the second technology was assessed.

The results of the assessment were captured in a TechSure template mind map [2]. This mind map was visible to the assessment team during the assessment on a
beamer screen, so quick feedback on the capture process could be given. In an evaluation several days after the second assessment, the team identified the following lessons learned:

- The team needs to investigate the support that the supplier offers for different versions of the software and the effort of migrating to new versions.
- The team identified that most of the sustainability risk is due to the monopoly position of the supplier. Hence, they need to conceive mitigation strategies to handle this risk.
- The team should further investigate the supplier’s IP position with respect to the technology.

With respect to the TechSuRe method itself, we learned the following lessons:

- The effort of learning a new technology should be part of the competence risk reasoning. If this effort is relatively small then the other indicators do not matter that much.
- The issue was raised how indicators related to standards should be treated when standards are not a major part of the technology, or are even irrelevant. For this, we opted to include a note in the description how to handle this special case.

With respect to the use of the mind manager tool, we learned the following lessons:

- The mind manager tool works well for documenting the valuation. It also provides good ways of hiding information making it nice for navigation and providing the ability to “zoom out” to provide an overview.
- Hint questions should be added to each indicator to improve understanding and to foster discussion among assessment participants. For example, for the supplier dependency on our company, we can ask the follow question: does the supplier go out of business if our company is no longer a customer?

B. ORM

In the second case study, the TechSuRe method was applied in the context of a technology evaluation of an Object Relationship Mapping (ORM) technology. In short, this technology aims to ease the modeling and transformation of domain models into traditional database models. The technology was not assessed for a particular product, but instead the evaluation tried to find out which products might benefit from this technology.

The technology evaluation was performed by a single person. To introduce the method to the technology assessor, we e-mailed the method description a week before the actual assessment took place. The TechSuRe assessment took place on a single day. In the morning, we presented the method and discussed some questions the technology assessor had with respect to the method. In the afternoon, the method was applied.

Once again, the template mind map was visible on a beamer screen and updated by us to capture the result of the TechSuRe assessment. We evaluated the TechSuRe assessment a week later with the technology assessor, whom identified the following lessons learned:

- The use of evolution scenarios to assess the technology evolution fitness uncovered two serious issues. This had a serious impact on the judgment of the technology assessor about the suitability of the technology.
- The method has a very good value for time ratio.

With respect to the method, we learned the following lessons:

- Our framework does not take into account that a supplier might be active in multiple markets, which might make the company too large to be taken over by one of their competitors in the particular market we are assessing the technology for. We changed the method description to include a broader perspective, which takes the total size of the company and the markets it operates in into account.
- The usage of a technology by a competitor can be both beneficial and detrimental to the impact the technology has on the evolution. The framework only highlighted the positive aspects of competitor usage, but missed out on the negative aspects like a competitor negatively influencing the evolution of a technology to frustrate its competitors. We updated the TechSuRe framework to take these aspects into account.
- Default or neutral values are also needed for the scenarios developed in the technology evolution fitness assessment. We updated the instructions for this part of the method accordingly.

With respect to the use of the mind map, we learned the following lesson:

- We added a completion tracker to each node, which tracks the extent to which the underlying factors have been assessed. In this fashion, the assessor can easily spot which factors are still “open” for assessment.

C. Cross-platform development toolchain

The last and third case study took place without the involvement of the original authors of the TechSuRE method, but was solely based on the materials supporting the method. In this case study, a R&D team had the task of deciding on a cross-platform development toolchain for a harmonized product line. The toolchain comprised both the programming and runtime environment, e.g. Java and VM or C/C++ with supporting libraries. The choice of this core
technology obviously has a major impact on the development organization, e.g., training of the development team and potentially requiring a migration of a large code base to the new toolchain.

In this case study, the method was not executed as prescribed by the authors. Instead, the mind map template was used to validate and extend the questionnaire to run the technology investigation of the cross-platform development toolchain. It acted as check-list for the completeness of the initial questionnaire. Sustainability of the technology was only one aspect of the evaluation. In the questionnaire, the sustainability factors were clustered into three categories: communities/market, lifecycle, and business. In the end, a revision of the initial questionnaire was created. After that step, the revision of the questionnaire was distributed to teams for each candidate technology for evaluation.

The assessors appreciated the detailed analysis of the sustainability factors found in TechSure. It provided good additional input for preparing the questionnaire that had a stronger focus on technical aspects before extending it based on the template with sustainability factors. Thus, TechSure can also be used in a checklist way for technology evaluations.

The assessment team identified two additional factors that were not originally in the TechSure method:

- **Interoperability:** On the one hand, interoperability is a quality of technology. However, interoperable technologies tend to have a more diverse market and thus the likelihood of sustainability problems is less pronounced. If one vendor goes out of business, there are still interoperable solutions that one can retreat to. For example, interoperability of UML tools nowadays is quite good, different tools can work on XMI exports. Modeling tools with a proprietary representation are not interoperable, thus if the project is no longer maintained, a sustainability risk becomes active.

- **Intellectual property:** If a technology is well protected by patents, then it usually does not have multiple vendors supporting it. Also, the risk for becoming sued is higher, even if using the technology in an intended way. On the other hand, a working license model can also ensure that the patent holding company has a regular revenue stream and thus is less prone to financial trouble. Overall, the IP situation has to be assessed and has an impact of the technologies’ sustainability.

### VI. CONCLUSION AND FUTURE WORK

This paper presented TechSuRe, a method for assessing the sustainability (risk) when choosing a new technology for a software intensive system. Using an elaborate reasoning framework, the method guides technology assessors in making sustainability assessments. The method has been improved in several iterations based on the feedback of the presented case studies and elaborated in the lessons learned.

One direction for future work is to try to further validate the method by retroactively applying it to old technologies and see whether the method makes the right predictions. A challenge for such work will be to ensure that only information that was available at the time is used in the assessment. Another direction for future work we foresee is to use the reasoning framework of TechSuRe as an inspiration source for defining a research agenda towards building more sustainable systems. Since many of the reasoning rules used lack a solid scientific basis, but are rather hypotheses that warrant further investigation. For example, we argue that a supplier’s technology commitment is mostly influenced by the market adaption and the supplier’s commitment. However, we lack concrete data to test this hypothesis. Another direction for future work is to expand the TechSuRe method to include a recommended way to report its findings. In particular, the interest is here how to communicate the results to stakeholders without a technical background.

### REFERENCES


