Symptom-Based Improvement Advice: A New Relevant-Focused Problem-Based Framework

Jan Pries-Heje[1], Jørn Johansen[2(✉)], and Morten Korsaa[2]

[1] Department of People and Technology, Roskilde University, Roskilde, Denmark
janph@ruc.dk
[2] Whitebox, Hørsholm, Denmark
{jj,mk}@whitebox.dk

Abstract. Classic maturity models require one or more trained assessors interacting with interviewees from the organization. This is a very rigorous and therefore costly process. In this paper we have developed the opposite; a relevant-focused and problem-based approach that can be used to generate improvement advice. We start out from the observation that trained assessors often observe symptoms when they enter an organization and find that there is an intuitive correlation with the given recommendations. We then developed a framework of symptoms and underlying problems using a technique called cognitive mapping. This framework was then used in a large organization to derive recommendations based on symptoms. This proof-of-concept seemed to work quite well. We present our framework and discuss how it can be implemented as an online tool. Further we identify several ways our approach can be further researched.

Keywords: Cognitive map · Process Improvement · Maturity · Improvement · CMMI

1 Introduction

It is of utmost importance to assess your organization's software process prior to launching an improvement. The rationale being that without an assessment of some kind, your intuition-based improvement plan would be at best groundless and ad-hoc and at worst counterproductive. Consequently, the work of most SPI professionals has always focused on assessment models and methods. Some of these models, such as SPICE [1, 2] and the Software Engineering Institute's CMMI [3], have gained widespread attention. Given the many different types of assessment, there is a predictable clamour over superiority. Some claim CMMI is better; others champion SPICE for specific areas such as the car industry [4] or in combination with agile development [5]. However, the effort required to perform an assessment is significant and is preventing many from establishing the required capability baseline that is needed to prioritize the required improvements.

We – the authors of this paper – have been doing assessments for many years. We have noticed organizations that flinched away from doing assessments because of the costs. Some have turned to self-assessments of maturity without much luck for good

M. Yilmaz et al. (Eds.): EuroSPI 2021, CCIS 1442, pp. 139–150, 2021.
https://doi.org/10.1007/978-3-030-85521-5_10

reasons like the ones described by Blanchette and Keeler [6, p. 23]. Another important observation done by the authors is an unfortunate consequence of asking professionals for compliance with a practice. The "why" gets lost! The good intention behind the practice is lost. The compliance with a practice is prevailing. The interviewees answer is "optimised" to demonstrate compliance because he seeks recognition from the assessor. But if the interviewee does not appreciate the reason to perform the practice, he is left with a feeling of being measured against a practice that does not make sense to him. This process does not appeal to his professionalism, and hence very little productive and creative discussions follow, that will encourage him to improve his way of working. The best assessors manage this communication during an on-site assessment to ensure motivation and encouragement. We have observed that the best interview technique is to ask if the interviewees experience a specific annoying symptom, and then discuss why this symptom is present. During that communication sequence, the assessor must cover all the related practices and assess what practice is poorly performed and causing the symptom.

This lead to a formulation of a research question "How can we give rigorous symptom-based improvement advice without an assessment?". The answer to this question will be highly relevant to many organizations that are either not big enough, motivated for, or do not have the time for a full-blown maturity assessment. In relation to the SPI manifesto [7] our research question is primarily related to part B (Business) and the recommendation to "use dynamic and adaptable models as needed". Secondarily our research question addresses part C (Change), "manage the organisational change", as an answer to the research question will give a starting point for an organizational change.

2 Experience

What if we could establish a system based on the authors experience, that could provide quick improvement recommendations? Perfection is obviously not the primary goal, but good enough to create value and then be improved based on systemic feedback. The baseline of experience is based on 25 years of performing assessments with the purpose of internal process improvement. In more than 500 cases, we have analyzed projects practices against CMMI/SPICE models to provide advice and recommendations to the project/organization including pointing to which practices that would bring the best performance improvement. It is important to notice that none of the assessments were done to prove compliance, but entirely for the purpose of internal process improvement. This means that the end result was always the motivation and guidance towards a better performance, as opposed to convince the assessor of compliance with a standard. Hence gold plating and cheating was not a problem. Approximately 95% of the recommendations have been acknowledged by the projects as relevant and following the recommendations has led to improved performance and proud professionals.

The responsibility of an assessor is to screen practices for weak areas and prioritize them. We use a reference model as guide to the assessment scope, and the specific/base practices as inspiration. The practices are examples of what good performers have done to achieve the goal of the practice.

The practitioner/interviewee can relate a practice in different ways. Here related to "The Four levels of teaching" [8]:

1. He does not know that he needs to do the practice. He does not see the benefit, and hence do not really care and have never invested in acquiring the competence. [The unconscious incompetence level]
2. He realizes that improving this competence will lead to a better performance. [Conscious incompetent level]
3. He is very aware of the benefit that this competence is providing. [Conscious competent level]
4. He is using the competence in his daily work as a natural thing and is really not aware of either doing it or the benefit it brings. Stopping doing it makes no sense at all. [Unconscious competent]
5. He is teaching others in the competence. [Expert level, (a fifth level often added to the original four levels)]

How an interviewee relates to a given practice, is very different dependent on his actual level. The authors believe that this is a fundamental weakness in the classical questionnaire based self-assessments.

To ask a relevant question you need to know what level the interviewee is at. What answer can you expect if you ask a toddler about when to shift to a higher gear on a bicycle? Or if you ask a seasoned cyclist on how to keep the balance? Or if you ask a professional about the benefit of not having father to run beside you with a stick to maintain your upright position. The analogy holds if you think of asking engineers about e.g., FMEA analysis. Some will be experts, and some will not even know what it is. And most will be in between, and may think that they are good at FMEA, but completely misunderstood the concept, and some may think they are terrible at FMEA, but in reality, they get all the benefits, but are aware of one little detail that is not performed optimal.

That describes an important part of the assessor's responsibility: To navigate in the interview and establishing the true competence of the interviewees to perform a practice and achieve the benefit. This is important for this paper's idea, because as assessors we learned something important: Everyone can relate to a symptom of bad performance:

- If you are unconscious incompetent, reflecting over the symptom may reveal just that, and make you conscious about the incompetence.
- If you are conscious incompetent, you can recognize the problem, and then you would like to improve.
- If you are conscious competent, you will point to underlying problems.
- If you are an unconscious competent, you most likely do not experience the symptom.
- If you are an expert, you will recognize the symptom from others performance.

Anyhow, we have learned as experienced assessors, that the interview flows freely, when we are discussing symptoms and underlying problems, and use our experience and the intelligence of the models to link the interviewees statements to model practices. And based on what we learn about the experienced problems, again using our experience and the intelligence in the models, we can provide relevant advice and recommendations.

Now the challenge of this paper is: Can we create an "AI" system, that first ask to some high-level symptoms. Then, based on the interviewees acknowledge of a given symptom, and his prioritization of the impact, can generate a new set of questions related

to the underlying symptoms. And based on his recognition and prioritization of these underlying problems – generate yet another set of questions to the underlying problems, and then link the analysis to specific practices in the model.

Second challenge is to link the analysis to relevant recommendations based on the catalogue of thousands of real-life recommendations from performed assessments. From analysis of those, we have found patterns, and they may relate to the underlying problems that the analysis revealed. If there is a reasonable correlation, then we can create a reasonable online alternative to a real live assessment or at least be a better alternative to practice-based self-assessment tools. We do expect that a live assessment with skilled assessors always will provide the best and most relevant recommendations. But considering the cost of a live assessment, a much cheaper online generation of recommendations may be valuable, which also can be used to follow improvements in teams, departments, and organizations.

3 Deriving a Framework

The research method we have chosen is called Design Science research (DSR). The core idea in this approach is to iteratively build and learn [9]. Hence, instead of trying to analyse every detail of a problem you build – or design – something with the best available knowledge and then you evaluate it in order to learn as much as possible. These evaluations of your design are called formative as the results are used to 'form' the solution. At the end of your project when you believe you are finished you do a so-called summative evaluation [10].

The first step in choosing an assessment strategy is to decide the purpose of your assessment. This is no easy task. Inevitably, you must consider your organization's history, culture, environment, and desire for change, as well as its vision for the future. Once you reach an understanding of the assessment's purpose, you are halfway to finding your assessment strategy. Your next task is to actually select your strategy from among the many options, knowing full well that many are largely incompatible.

From a literature study of assessment approaches [cf. 11] we identified two dimensions that are important for distinguishing between assessment strategies.

- Rigor vs. relevance. At one extreme, an assessment strategy may have a very strict approach to collecting data and using it to draw conclusions. We call this a rigor-based strategy. Typically, a rigorous method follows rules to achieve systematic and defensible results that are even considered objective in some cases. The opposite of a rigor-based strategy is a relevance-based strategy. Here the strategy focus on eliciting results relevant for one or more of the stakeholders. Or said in another way, a relevance-based strategy may not lead to an assessment that is full or complete or rigorous, but it will lead to results and conclusions that concern the involved organizational actors.
- Model-based vs. problem-based. A model-based method puts its faith in maturity models such as CMMI or SPICE. More generally, the model-based strategy focuses on models of best practices and, consequently, models of general software-process problems. A problem-based assessment strategy focuses on what software practitioners and their managers perceive as problems in the software process.

These two dimensions form a framework as shown in Fig. 1.

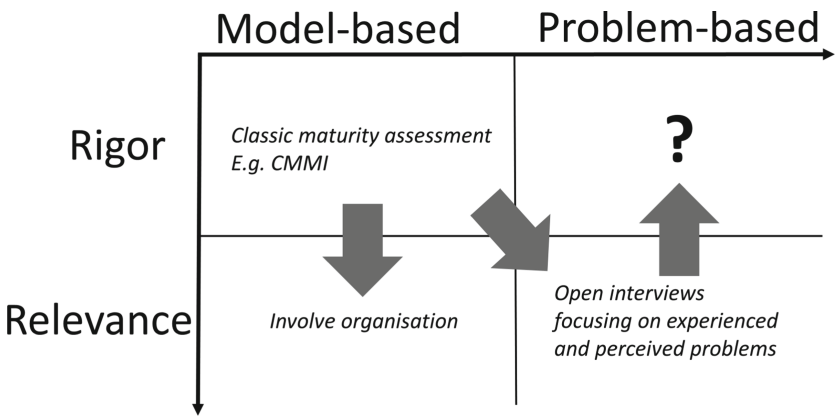

Fig. 1. A framework where the question mark shows where we are aiming

In Fig. 1 in the upper left corner, we have a classic maturity assessment performed with a rigorous model and method such as CMMI. One challenge is that it is often costly as mentioned above. Another that it may not get enough insight into the developing organization. The first arrow on the left in Fig. 1 illustrates a road often taken namely to involve the organization e.g., by having internal assessors participate in the assessment. The second arrow from the left points to the opposite approach where you do not have the categories of a model such as the CMMI but instead rely on open interviews focusing on what the people in the organization experience and perceive as problems.

The last arrow to the right in Fig. 1 points to what we aim at with our research question, a non-costly approach with the rigor of a classic assessment but instead taking the starting point in the problems or symptoms of practice.

To answer the question mark in Fig. 1 we applied coloured cognitive maps (CCM) originally developed by Colin Eden [12]. In 1988 he came up with cognitive maps of problems or constructs as he calls them where the link between two problem constructs is in the form of an arrow to show the nature of the linkage; "an arrow out of a construct shows a consequence and an arrow into a construct an explanation" [12, p. 5]. Some years later Colin Eden and Fran Ackerman [13] developed cognitive or causal maps into a technique [14] that could be used for making strategy. Finally, Venable [15] refined cognitive maps into coloured cognitive maps that can be used for creating a design of a solution. His idea was that each problem should be formulated with its opposing node. E.g., 'high employee turnover' has the opposing node 'low employee turnover'. And when you switch around a complete cognitive map – from the original nodes to the opposing nodes - you will end up with a map of potential design solutions.

4 The Framework We Developed

The idea we pursued was that business problems are dependent on a set of patterns and relationships between symptoms and underlying problems and symptoms leading to the

experienced business problems. The authors have, based on the experience from 600+ assessments, first defined the most commonly recognized symptoms. Then listed the underlying problems that is causing the symptom. Same thing on level deeper: What is the underlying problem to this underlying problem that causes the symptom, and then finally mapping these to one or more practices in the CMMI maturity model [3].

We did the same analysis moving "up" from a given symptom, asking what business problems that the symptom is causing.

The authors acknowledge the lack of research behind this part of the process but hope that the analysis will be validated by experiments, following the principles of clinical research.

So, the logical model behind is a five-level root-cause analysis based on the 5-times-why techniques [cf. 16], but mapped as a complex many to many networks, not assuming that there is only one problem causing another problem. An example of a symptom – We often have reflux (things we need to redo and improve) – and the map can be found in Fig. 2.

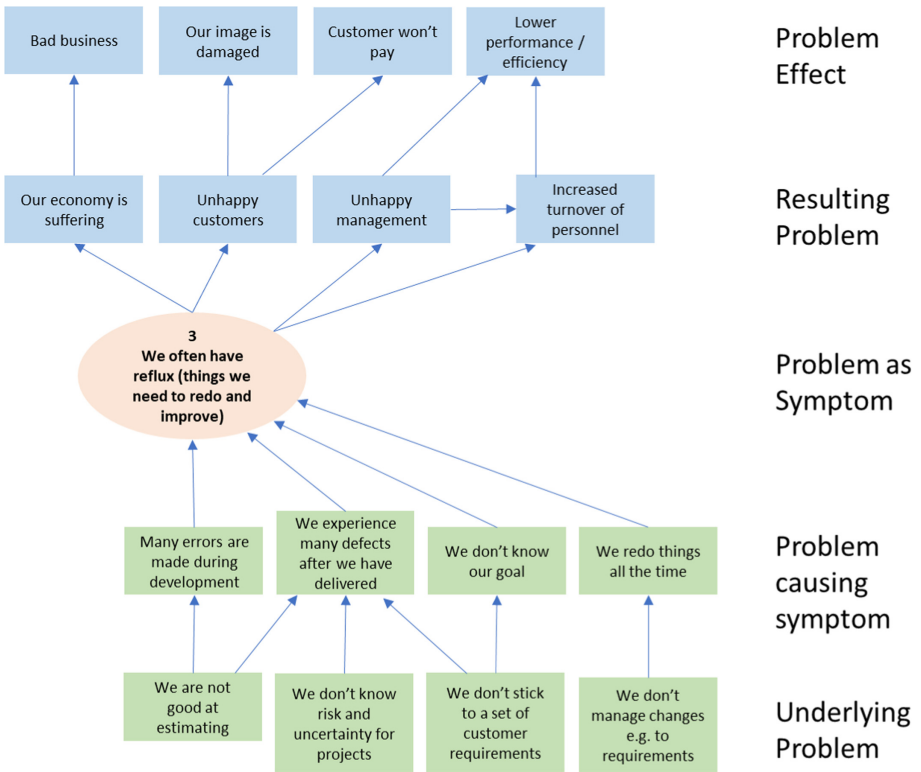

Fig. 2. Example of the five levels for symptom #3

The first step is to identify the symptoms which relates to worst problems. For doing that we developed a questionnaire and an online response and some simple overall recommendations.

We developed a questionnaire with 32 symptoms in five categories:

1. In Control
2. Knowing what to develop and deliver
3. Project execution capability
4. Project overview
5. Quality in tasks and daily work

In Fig. 3 we have shown all the symptoms from the five categories.

The online system will, when it is developed, generate the relevant questions based on the network, by moving up and down in the network, and asking if this "problem a" is part of the cause to "problem b". In that way we identify the most relevant related problems leading to weakly performed practices, where the impact of poor performance is acknowledged by the interviewee.

To try the concept out, we did a manually interaction and interviews to identify the related problems – leading to the relevant weak practices, causing the problems.

We are working on using the interviewees prioritization/impact assessment of each problem to score the impact of the practices. If more problems are pointing at the same practice, it scores more. If more interviewees from the same project is pointing at the same problems, it scores more. In the end, the score determines the prioritizations between the practices, and hence determines the category of recommendations. It will strengthen the result and the recommendations.

In the following section we present the result of the case study using the concept.

5 Case Study

A case study is a research strategy that investigates a phenomenon within its real-life context. Every assessment we the authors have made can be seen as a kind of case study [17]. However, in this paper we decided to apply the framework we had designed using design science research as our method [9] and having done several iterations. However, Venable et al. [10] recommend that you do as 'naturalistic' an evaluation as possible meaning that you have real users, in the real context with the real problem. Hence, we have undertaken a proof-of-concept case study where we test the idea in order to demonstrate its feasibility [18].

The framework was used in a large organization to derive recommendations based on symptoms. This proof-of-concept seemed to work quite well.

The questionnaire described in Sect. 4 was given to a manager and a leading developer in Big Embedded Software Maker (pseudonym chosen for anonymity). The manager answered the questionnaire in the scope and situation of his department. The answer identified four symptoms as totally agreed (symptom #7, #15, #21 and # 32), 9 as partly agreed, 1 as neither nor, 15 as partly disagree and 3 as totally disagree.

Nr	Statement	Totally agree	Partly agree	Neither nor	Partly disagree	Totally disagree
	In control					
1	We are not able to account for the time used by individuals on a given activity					
2	We rarely keep our budgets					
3	We often have rework (things which we need to redo or correct)					
4	We experience difficulties when working together with others					
5	We experience problems we should have foreseen					
6	We find it difficult to fix defects in the things we have delivered					
7	Our key employees is a limitation to our growth					
	Knowing what to develop and deliver					
8	We do not know who is influencing a project and its results					
9	We have no direct access to the user/customer during the project					
10	We do not distinguish clearly between needs, requirements, and specifications					
11	It's difficult to make clear decisions when the conditions changes					
12	We are not managing changes in needs, requirements, and specifications					
	Project execution capability					
13	We do not find the defects before the customers					
14	We find many defects during integration					
15	Documentation are not updated continuously (specifications, architecture, design, …)					
16	We lack helping guidelines (manuals, examples, templates, …)					
17	If we find a defect it's difficult to see the consequences and get an overview of the work that follows					
18	We do not know the consequences for other customers/users when we fix a defect					
	Project overview					
19	We often correct the same mistake twice					
20	We do not know how much rework we have					
21	People are often disturbed by problems not related to their current					
22	We do not know our performance on the different types of tasks					
23	We do not utilize resources effectively across projects/teams					
24	The employees shifts between several task/projects through the					
25	We find it difficult to learn from our mistakes					
26	We cannot show documentation for the rationale behind the important decisions					
27	We often experience that our tool support is causing problems					
	Quality in tasks and daily work					
28	Employees experience receiving bad "deliveries" from colleagues					
29	No one cares about how we work (e.g. how an review is conducted)					
30	No one knows if the work instructions/process descriptions/tools are wisely used					
31	Our customers/users are often dissatisfied					
32	We have no plan or strategy for competence development					

Fig. 3. The content of the five categories – as a questionnaire of symptoms

We focused on the 4 totally agreed symptoms, and for each of the symptoms asked about the relevance of the related problems causing the symptom and the underlining problems related to the problems causing the symptom.

We take an example. One of the totally agreed symptoms was #21: "People are often delayed by things that suddenly appear". In Fig. 4 we have shown the map for this symptom.

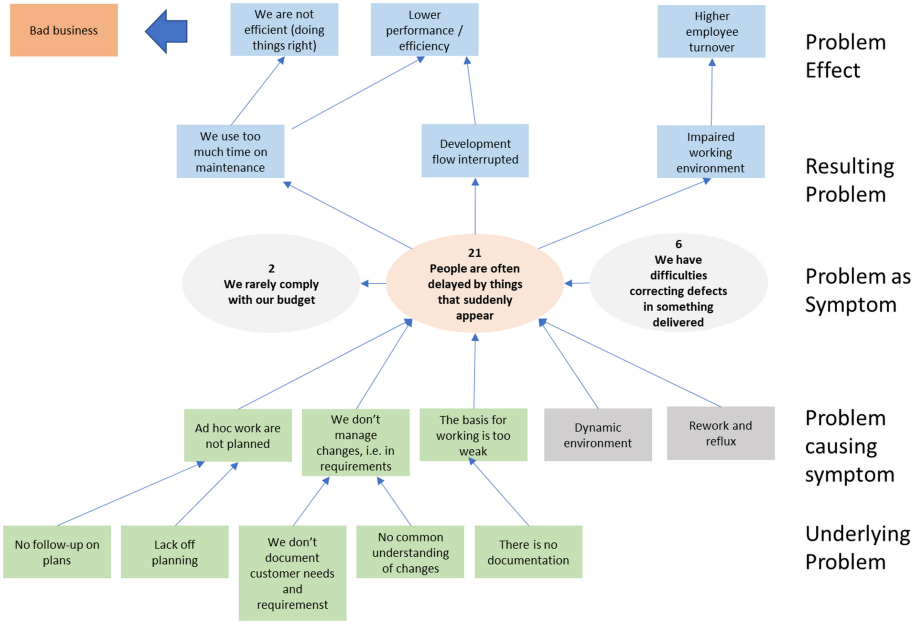

Fig. 4. The map for symptom #21 - "people are often delayed by things that suddenly appear"

As an example we have shown the two steps of dialogue we had with one of our interviewees for symptom #21:

1. Evaluate the problems causing the symptom and indicate if it is very serious, serious, less serious, not serious, or Not applicable.

#21: People are often delayed by things that suddenly appears	a. Ad hoc work are not planned
Answer: a. less serious **b. and c. serious**.	**b. We don't manage changes, i.e., in requirements**
	c. The basis for working is too weak

b. and c. were chosen for the second dialogue.

2. Indicate which of the underlying problems is the most relevant.
b. We don't manage changes, i.e., in requirements.

1. Is it because the customer's needs, requirements and specifications not are maintained?
2. Or because there is no common understanding of change?
3. Or something completely different?

Here the underlying problem 1. was chosen with the comment:" Yes and no, as we typically work with something that has never been made before, requirements change along the way, when you have become wiser – and not always updated."
c. The basis on which there is too little work.

1. Is it due to lack or too little documentation?
2. Or something completely different?

Here the underlying problem 1. was chosen.

Finally, we had a discussion with the interviewees on which resulting problems and problems effects he found the most relevant.

Here the example interviewee explained that the most relevant resulting problem was Development flow interrupted and the most relevant problem effect was Lower performance/efficiency. In Fig. 5 we have indicated the answers received in the summarised example.

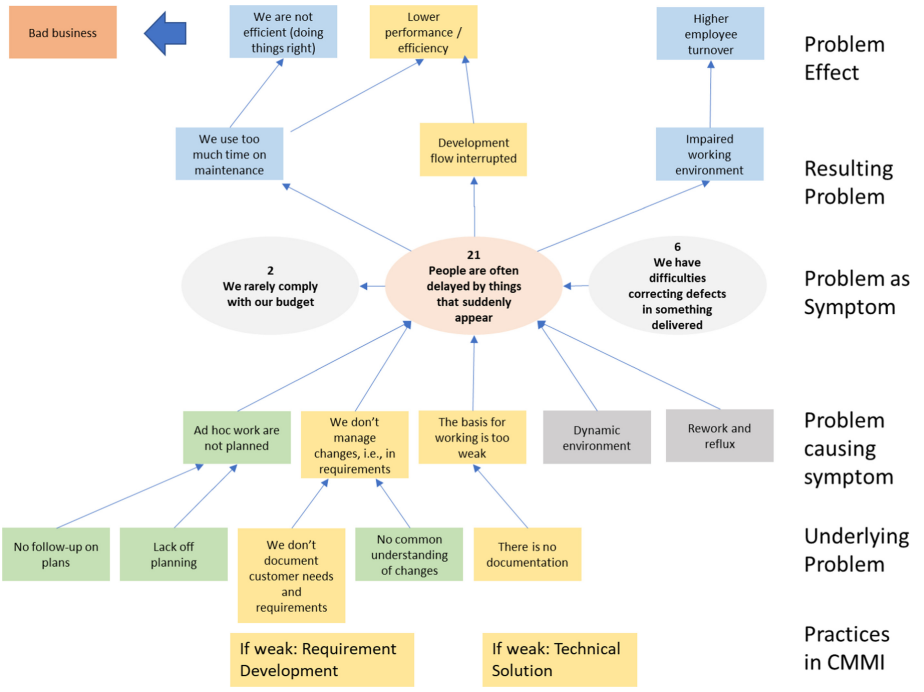

Fig. 5. The answers in yellow from one chosen interview example (Color figure online)

From the underlying problems to practices in maturity models like CMMI, there is a clear connection between "We don't document customer needs and requirements" to the practices in the process Requirement Development, and from "There is no documentation" to the practices in the process Technical Solution.

Now we have a full "picture", where the yellow boxes in Fig. 5 indicate the connection from weak practices to problem effects.

The same process and dialogues took place for the other 3 symptoms, and we found similarly "pictures" for those symptoms.

When we discussed the picture of e.g., Symptom #21 the manager found it relevant and valuable, especially to see the connection to the effect of problem effect.

What we found essential was the easiness in the discussions. As an assessor it is always important to ensure dialogue responsibility (both the interviewer and the interviewee understand each other), which we found was in place here. It is much easier to discuss symptoms and problems first, and then the needed practices to improve the situation, than focus on what cleverness and good practices are.

6 Discussion and Conclusion

One important thing to consider is whether we need to pursue not only the symptoms which were totally agreed but also include symptoms that were partially agreed. One mechanism we have looked at is to trace backward from practice level (e.g., for Symptom #21 If week Requirement Development to other symptoms, which also refer to this practice, to investigate if these "related" symptoms were partially agreed, which could strengthen the argumentation and the result. For symptom #21 this will be the situation for symptom #3 Reflux.

Another possibility is to include more than one person in answering the questionnaire described in Sect. 4, and let symptoms chosen as fully agreed by both respondents have a larger weight as well as symptoms chosen as fully agreed by one and as partial agreed by the other. Then it will be possible to weigh the symptoms and produce a stronger result and recommendations. This would be more alike what we do in a normal assessment.

We have now developed and presented a "way" to go from symptoms to recommendations. Further, we have evaluated our design in several formative assessments and a summative evaluation [10]. Further research will be to build a "map" on the relations between the symptoms based on the underlying common connections to practices in maturity models. This will make it possible to build a strong dialogue with the respondents for identifying the most relevant connections from problem effect to practice – as illustrated in Figs. 2, 4 and 5.

Further work on our framework could possibly also include a self-learning functionality based on two functions:

1. The feedback from the users, after deployment, will tell if the relations are valid.
2. If no underlying problems is relevant to a symptom, the user is invited to type in what he experiences as the underlying problem. When more interviewees report the same underlying problem, the network has become smarter. This, however, requires a manual analysis and update process, that remains the responsibility of the model owner.

References

1. Mas, A., et al.: Software process improvement and capability determination. J. Comput. Stand. Interfaces (2017). Elsevier
2. Dorling, A.: SPICE: software process improvement and capability determination. Softw. Qual. J. **2**(4), 209–224 (1993)
3. Team CP, CMMI for Development, Version 1.3, Pittsburgh, 2010. Software Engineering Institute, Carnegie Melon University, February 2018
4. Macher, G., Much, A., Riel, A., Messnarz, R., Kreiner, C.: Automotive SPICE, safety and cybersecurity integration. In: Tonetta, S., Schoitsch, E., Bitsch, F. (eds.) SAFECOMP 2017. LNCS, vol. 10489, pp. 273–285. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66284-8_23
5. Komiyama, T., Konno, S., Watanabe, T., Matsui, S., Kase, M., Igarashi, I.: Improvement of agile software development process based on automotive SPICE: a case study. In: Walker, A., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2019. CCIS, vol. 1060, pp. 518–531. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-28005-5_40
6. Blanchette Jr., S., Keeler, K.L.: Self-assessment and the CMMI-AM-A Guide for Government Program Managers. Software Engineering Institute, Carnegie-Mellon University, Pittsburgh (2005)
7. Pries-Heje, J., Johansen, J.: SPI Manifesto, Version 1.2. In: European System & Software Process Improvement and Innovation (2010)
8. Broadwell, M.M.: Teaching for learning (XVI). The Gospel Guardian **20**(41), 1–3 (1969)
9. Hevner, A.R.: A three cycle view of design science research. Scand. J. Inf. Syst. **19**(2), 4 (2007)
10. Venable, J., Pries-Heje, J., Baskerville, R.: FEDS: a framework for evaluation in design science research. Eur. J. Inf. Syst. **25**(1), 77–89 (2016)
11. Nielsen, P.A., Pries-Heje, J.: A framework for selecting assessment strategy. In: Improving Software Organizations: From Principles to Practice. Addison-Wesley, Boston (2002)
12. Eden, C.: Cognitive mapping. Eur. J. Oper. Res. **36**(1), 1–13 (1988)
13. Eden, C., Ackerman, F.: Making strategy. In: The Journey of Strategic Management. Sage Publications, Londres (1998)
14. Ackerman, F., Eden, C., Cropper, S.: Getting started with cognitive mapping: tutorial notes. Strathclyde: Strategic Decision Support Research Unit, vol. 18. Strathclyde University (2008). Retrieved Nov 1992
15. Venable, J.R.: Using coloured cognitive mapping (CCM) for design science research. In: Tremblay, M.C., VanderMeer, D., Rothenberger, M., Gupta, A., Yoon, V. (eds.) DESRIST 2014. LNCS, vol. 8463, pp. 345–359. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06701-8_25
16. Liker, J.: The Toyota Way. Esensi (2004)
17. Yin, R.K.: Case Study Research: Design and Methods, Applied Social Research. Methods Series, vol. 5 (1994)
18. Kendig, C.E.: What is proof of concept research and how does it generate epistemic and ethical categories for future scientific practice? Sci. Eng. Ethics **22**(3), 735–753 (2016)