Symptom-based improvement recommendations

Pries-Heje, Jan; Johansen, Jørn; Korsaa, Morten

Published in:
Journal of Software: Evolution and Process

DOI:
10.1002/smr.2375

Publication date:
2023

Document Version
Early version, also known as pre-print

Citation for published version (APA):
Pries-Heje, J., Johansen, J., & Korsaa, M. (2023). Symptom-based improvement recommendations. *Journal of Software: Evolution and Process*, *35*(8), Article e2375. https://doi.org/10.1002/smr.2375

Symptom-based improvement recommendations

Jan Pries-Heje[1][0000-0003-2219-9332], Jørn Johansen[2][0000-0003-1368-3640] and Morten Korsaa[2][0000-0003-4521-454X]

[1] Dept. of people and Technology, Roskilde University, Denmark
[2] Whitebox, Hørsholm, Denmark
janph@ruc.dk, jj@whitebox.dk, mk@whitebox.dk

Abstract. When an experienced maturity assessor enters a company, there are certain characteristic symptoms that are noticeable that reveals the maturity of the company even before the assessment. For this paper we identified a list of 32 characteristic symptoms to notice, generated by two experienced assessors who had undertaken maturity assessments in more than 300 companies. We then use cognitive mapping and the 'five whys' technique to look beyond the symptoms and reveal the underlying problems or causes of the problems. Following that we evaluate our findings through the design and evaluation of a web-based tool where users can score statements based on a formulation of the symptoms. This enables us to recommend to users the areas where they probably need to improve. We designed the tool in three learning cycles of design evaluation and ended up in a summative evaluation where we compared the outcome of using the website tool with a CMMI maturity assessment. We conclude that a systematic quest for symptoms coupled with scoring statements based on the symptoms can point to improvement areas. However, we conclude that doing so is no substitute for a maturity assessment; the scoring of statements cannot reveal the maturity of the organization but it can quickly and easily point in a useful direction and provide recommendations for going in that direction.

Keywords: cognitive map, process improvement, maturity, improvement, CMMI

1 Introduction

Imagine that you are a manager in a product development company or department, in distress over some bad performance (too late, too expensive, poor quality...). Somehow it is your responsibility that you are not performing, but where is the real problem? Which practices needs your attention? What should you do to improve?

Start looking, and you will find an overwhelming number of processes, problems, causes of problems, symptoms of problems, tools, practices, people, organizational structures. Each of them could potentially be the problem. And they all seem to re-late to each other in a complex set of ways.

Maturity models such as the Capability Maturity Model Integrated - CMMI [1, 2] have been around for diagnosing problems and recommending improvements for more than 25 years and are still used every day. Most recently an ISO standard [3] for how

to diagnose – or assess as it is called – was published along with a maturity model. Studies have shown [4-6] that increasing your maturity as an IT or Software developing company bring about real benefits such as higher quality of your products, less rework, and faster development of new products.

Today, success is best demonstrated by those industries that have embraced the maturity model concepts to increase quality and performance. The European automotive sector is probably the best example, having made many companies and industries envious when watching the benefits they show based on a newer maturity model , Automotive SPICE, on which Falcini et al. [7] says: "Because of Automotive SPICE's pervasive adoption and holistic coverage of automotive-software development, it's the appropriate reference for systematically analysing deep learning for automotive software engineering and for promoting a mature, harmonized methodology for deep learning"

However, applying maturity models often comes at a high cost. When a manager in a company decides to follow this approach, just deciding on the most relevant maturity model can be very stressful, as there are quite many that all appear to be fit. Further, the organisation needs some training, and often an external assessor has to be involved. The effort required to plan and perform the assessment is normally much more than can be handled without schedule consequences for the involved employees and projects. Hence, it takes the involvement of top management to prioritize and accept the immediate drop in performance incurred. While the return of investment may be worth it, the investment is blocking many organisations from establishing the baseline to improve from. This has recently been found to be a major reason for the standstill in industry improvement as reported in Johansen & Andersen [8].

The authors of this paper aimed to identify an alternative approach that are faster and leaner and can bring a (top) manager useful recommendations about which to focus at. Less precision in the recommendations is expected – and acceptable - when making shortcuts to a well-established assessment process. However, precision may not be of utmost importance when you have many practices that needs improvement. As long as the recommendations are fast and targeted at relevant areas the approach will provide value. Success is when a manager with minimal effort is directed towards the most important practices to improve.

Hence, that is the research question we try to address; How can we design a simple and fast approach to establish recommendations for an organisation?

In answering this research question we have taken our starting point in the more than 600 assessments that two of the authors have carried out primarily using the CMMI model (ref). Each assessment generated on 20 pages of notes from questions and answers related to the performance of all the processes in the CMMI model [1]. In a prestudy to this paper we established through coding [9] of the interview notes that there is a relatively limited number of symptoms that is expressed by the interviewees as being of real importance in their organisations. There are also a limited number of relevant relationships between the symptoms, the experienced problems, and the causes of the problems. Hence, in this paper we start out from a list of 32 symptoms that we identified from coding the interview notes and assessment reports. The 32 symptoms are:

1 We are not able to account for the time spent by individuals on a given activity
2 We rarely keep to 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 are a limitation to our growth
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 is difficult to make clear decisions when conditions change
12 We are not managing changes in needs, requirements and specifications
13 We do not find defects before the customers do
14 We find many defects during integration
15 Documentation is not kept up to date (specifications, architecture, design, etc.)
16 We lack helpful guidelines (manuals, examples, templates, etc.)
17 If we find a defect, it is 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
19 We often correct the same mistake twice
20 We do not know how much rework we have
21 People are often annoyed by problems not related to their current task
22 We do not know our performance on the different types of tasks
23 We do not utilize resources effectively across projects/teams
24 Employees switch between several task/projects through the week
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 find that our tool support is causing problems
28 Employees experience receiving ‘bad deliveries’ from colleagues
29 No one cares about how we work (e.g. how a 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

After having identified this list we take each symptom and ask two things: (1) What are the causes of this symptom? That is to say, the problem underlying it; and (2) What does it cause or lead to or affect?

An example symptom could be that the documentation for a product is not updated. That could be caused by lack of time for preparing documentation, and that again could be caused by a very tight development schedule or budget for product development. Furthermore, when documentation is not updated it could lead to difficulties in maintaining and further developing the product, and that again could lead to the problem that it becomes excessively costly to maintain and further develop the product.

This example has five levels of problems, that is, a kind of problem hierarchy. An well-known Japanese improvement technique called 'five whys' operates with precisely five levels, the point being that you should never go with the first symptom of a problem but instead look for the root cause(s) by asking 'why' five times. The Japanese car company Toyota developed the five whys technique in the 1930s. It became popular in the 1970s, and Toyota still uses it to solve problems today [10]. The reasoning behind the five whys techniques is that you need to search for the root cause of problems instead of trying to cope with superficial symptoms of problems caused by deeper and underlying issues. The former executive vice president of Toyota Taiichi Ohno uses an example of a welding machine to explain the importance of asking why five times [11]. The example goes: (1) Why did the robot stop? Because a circuit has overloaded, causing a fuse to blow. (2) Why? Because there was insufficient lubrication on the bearings, so they locked up. (3) Why? Because the oil pump on the robot is not circulating sufficient oil. (4) Why? Because the pump intake is clogged with metal shavings. (5) Why? Because there is no filter on the pump. Hence, in this example [cited from 11] you should not change the fuse or the circuit but instead solve the underlying root cause of the missing filter.

The research question we aim at answering in this paper is: 'How can we design a simple and fast approach to establish recommendations for an organisation?' In order to answer that we use a mapping of the symptomatic problems, causes and effects, and the relationships identified between them, to design a tool that can help determine what the most urgent improvement areas are in a given organisation.

In our research we have used the CMMI [1] maturity model as our basis for discussions. This choice was based on the fact that two of the authors are very experienced in using this model. However, it could be equally relevant for use with any of the other nearly 200 maturity models [12] that can be found, such as ISO/IEC 15504 [13] or Automotive SPICE [14], and the addressed practice capabilities in these models.

2 Existing research on problems

A problem can be defined as a perceived difference between what is and what ought to be [15]. There can be many aspects of a problem [16]. It could be the awareness of a gap, a desire or a need. On the other hand, it could also be that something is undesirable and therefore implies the imperative for change. A third aspect might be that it is difficult as opposed to trivial. A fourth aspect, that it is solvable as opposed to impossible to solve. And finally, there can be different perceptions by different stakeholders of how problematic something is. One stakeholder might see it as undesirable not to have updated documentation, whereas another might see it as no problem at all.

In the literature you find many problem analyses and problem-solving techniques. A well-known method by Peter Checkland [17] [18] [19] is called Soft Systems Methodology (SSM). Checkland had experience as a consultant for big international companies such as Shell before coming back to academia as a professor in systems thinking. He distinguishes between hard and soft problems. A hard problem is a well-formed problem that can be solved with well-known engineering techniques. The problem presents

itself so that it is easy to see what type of problem it is. A soft problem, on the other hand, may have many aspects, many humans involved and many different stakeholder perspectives. Thus, it needs work and discussion to understand the problem – if it is indeed a problem!

Checkland has two important points that we will use here. First, it pays to distinguish between the real world and systems thinking at a meta-level about the real world. Second, it pays to produce models of purposeful activity in the real situation and use the models as devices to explore the situations and to structure a discussion.

Some problems can be defined as being 'wicked' [20, 21] – more or less [22] – meaning that you cannot solve the problem unless you have some knowledge that you can only obtain by solving the problem; the problem cannot be understood until after the formulation of a solution. Hence, the only viable strategy is to start solving the problem and learning in the process. Furthermore, wicked problems can be considered to be a symptom of another problem. They are linked together.

Years later Snowden and Boone [23, 24] presented a framework for sense-making to be used by leaders for decision-making where they look at the relationship between cause and effect. If the relationship between cause and effect is known, it is a 'simple problem' to which we can apply best practice. If the relationship is potentially knowable – for example, through hard work by experts – then it is a complicated problem. And if the problem is wicked and thus the relationship between cause and effect only retrospectively coherent, then it is a complex problem.

Looking at the relationships between problems and the causal relationships, Colin Eden [25-27] 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 link; 'an arrow out of a construct shows a consequence and an arrow into a construct an explanation' [25, p. 5]. Eden for example used cognitive maps to identify implications of a decision [25, p. 4] thereby being able to make better decisions and carry out – as he call it – "reflective problem solving". Eden also used to map the nature of decision making [26, p. 266] taking into account different perspectives of different stakeholders. And Eden with three co-authors studied the effect of design changes and delays and showed by using cognitive maps that changes caused delays in a large commercial project [28]. Some years later Eden and Ackerman [29] developed cognitive or causal maps into a technique [30] that could be used for making strategy. Finally, Venable [31] refined cognitive maps into coloured cognitive maps that can be used for creating a design for a solution. His idea was that each problem should be formulated with its opposing node. For example, 'high employee turnover' has the opposing node 'low employee turnover'. When you switch around a whole cognitive map, from the original nodes to the opposing nodes, you will end up with a potential design solution. In the example given, that means that 'lower the employee turnover' could be seen as a potential design solution for the problem of 'high employee turnover'.

3 Research method

To answer our research question ‘How can we design a simple and fast approach to establish recommendations for an organisation?’ Thereby being able to answer for an organisation; ‘what are the most urgent improvement areas in your company?’, we decided to apply DSR – Design Science Research [32]. DSR is a research approach where you build something and then learn from it (when evaluating). Thus, in order to answer our research question, we decided to build cognitive maps showing the links and relationships between problems and symptoms, causes and effects, to better understand what the most urgent improvement areas in a company might be. And we decided not to do it for a specific company but instead do it at a systems-oriented meta-level in order to answer our more generic research question.

A major reason for choosing DSR as our research methodology is that it combines the need for practical relevance and utility. DSR emphasizes that a design should address a need or a problem and at the same time should ‘stand on the shoulders’ of existing research in the problem area [32]. Besides having a ‘relevance iteration cycle’ where you start by identifying a need or a problem, you also have a ‘rigor iteration cycle’ where you identify all relevant academic literature; what do we actually know by now? The artefact that you are building in order to learn can be a product artefact or a process [33].

Hence, we developed causal cognitive maps. We started out from the 32 symptoms that we had seen in companies and asked what can this cause (= consequences) and what is causing this? As mentioned above, we decided to use the five whys technique, so we developed each symptom at five levels, typically starting with the symptom in the middle of the map and then eliciting two levels of consequences and two levels of causes.

What data should we use for creating the maps? Here we took advantage of the more than 600 assessments in more than 300 companies that the group of authors have together carried out. That has given us extensive knowledge of how things are related. So we simply used cognitive mapping techniques to make explicit what was in our minds. At first, we split the symptoms into three groups and mapped a group each. Then, to avoid bias and give some inter-coder reliability, we swapped the maps around among us until all three authors agreed on the links and relationships. We decided to represent only the most important link(s) – no more than three of them. A symptom can be caused by many things, but we decided to prioritize the causes and only represent the most important ones. In doing so we are following the principle of organizational learning from the SPI Manifesto [34].

This paper primarily supports the following principles in the SPI Manifesto: Create a learning organization; Support the organization’s vision and business objectives; Use dynamic and adaptable models as needed – in the sense of bringing insight to the principles.

4 Symptoms observed

In this paper we start out from a list of 32 characteristic symptoms prepared by two of the authors of this paper, experienced assessors who have undertaken maturity assessments in more than 300 companies. They were used in the following way.

Over the past 25 years, as many as 600 assessments were performed in over 300 companies. During this work, the assessors became increasingly adept at picking up signals related to how 'clever' the company is at developing new products or delivering projects for customers. It even came to the assessors being able to guess the maturity after looking at some of the main documents and development model and being welcomed at reception.

Last year, the assessors started to identify the most common symptoms. After several brainstorms and discussions, they ended up with 32 symptoms, which were formulated as statements of the symptoms. Examples of symptoms were:

- We cannot tell how much effort an individual has expended on an activity (no. 1)
- We do not know who and how many have a say in the project and the results of the project (no. 8)
- Unfortunately, we do not find defects until the product is in service with the customer or end user (no. 13)
- We are often correcting the same mistake again and again (no. 19)
- Employees experience 'bad deliveries' from colleagues (no. 28)

During the latest assessments, we identified how often we discussed problems during assessment interviews and who they linked to the symptoms. We also consulted several of the notes form the many performed assessments, to qualify the conclusion on the 32 symptoms, and the link to discussed problems.

A few of many examples on remarks during an assessment – here form the from the latest assessment:

- Symptom #2 and related quality problem: "We have had some problems with project cost and reaching the right quality".
- Symptom #7 and #24 and related resource management problems: "We currently have too many parallel things going on – using the same key people", and "We are challenged by people being taken off projects due to customer projects".
- Problem related to several symptoms: "I See some bottleneck problems, and problems on prioritization".

The 32 symptoms were grouped into five categories, based on the category or type of symptoms (grounded in our coding [9]). The five examples (no. 1, 8, 13, 19 and 28) above are one from each of the following five categories:

1. To be in control of the projects across the organization

2. Knowing what do develop and deliver
3. Projects having staff with the right competences to run projects
4. Having project insight and status
5. Quality in work and work products

Having these symptoms defined, we had to qualify them and find a way to make this operational.

5 Cognitive maps

To obtain a better understanding of the symptoms, we decided to apply cognitive maps. We started out from a symptom. Then we asked 'what is causing this?' And then we asked again 'what is the cause of the cause?', thereby identifying the underlying problem. We then went on to ask 'what is the effect or result of the symptom?' And finally, 'what is the effect of the effect?' For some of the problems we could probably have continued further back to an even more deeply underlying problem or further forward to an effect of an effect of an effect. However, we had decided to apply the five whys heuristic so we ended our mapping of each symptom with a map that had five layers.

We did this mapping for all 32 symptoms. Furthermore, we circulated the maps among the three authors thereby neutralizing any bias that any one of us may have had. We also had some discussions about certain problems; is this a cause or an effect? That was not always easy to answer.

An interesting observation that we made while going through the symptoms one by one was that some (causal) problems or effects started to reappear. We noted that and discuss it later in this paper.

In Figure 1 you can see an example where we have mapped symptom no. 6: 'We have difficulties correcting defects in something delivered.'

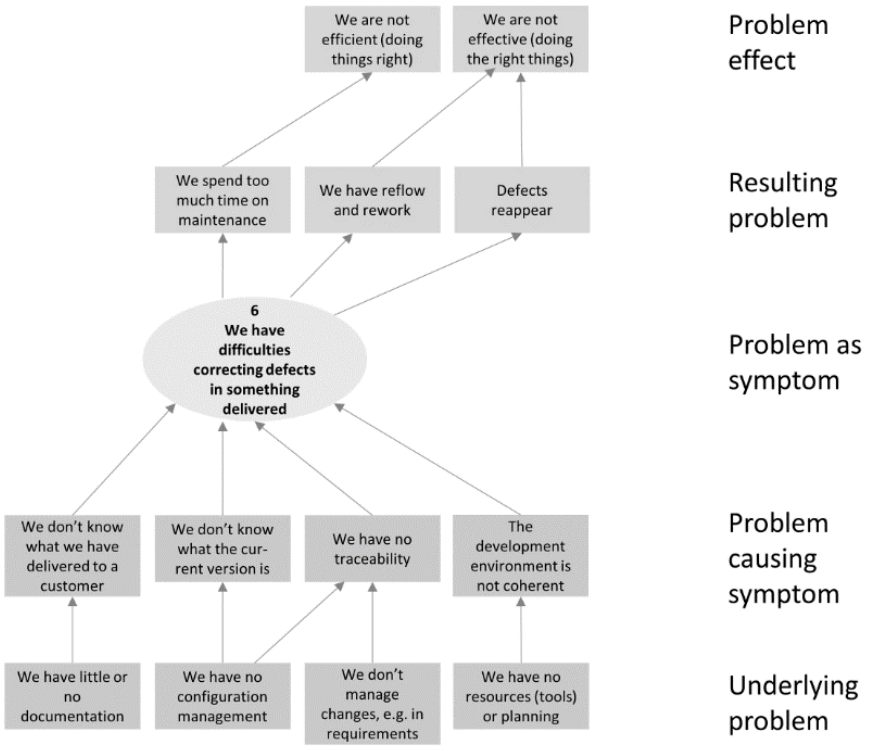

Fig. 1. An example of a cognitive map – here, of symptom no. 6. As you can see, the map has five levels of problems corresponding to the five whys heuristic that we have used.

One 'causality track' that we find in Figure 1 is the following: We have no configuration management => We don't know what the current version is => We have difficulties correcting defects in something delivered => We spend too much time on maintenance => We are not efficient (doing things right).

Another causality track found in Figure 1 is: We don't manage changes, e.g., in requirements => We have no traceability => We have difficulties correcting defects in something delivered => We have reflow and rework => We are not effective (doing the right things).

When we were eliciting and creating the cognitive maps, we also found that some symptoms are related to other symptoms. That was easily seen when two symptoms resulted in (or caused) the same effect or when two symptoms were caused by the same underlying problem(s). In Figure 2 we have shown the same symptom no. 6 as we presented in Figure 1 but now with two closely related symptoms represented as well.

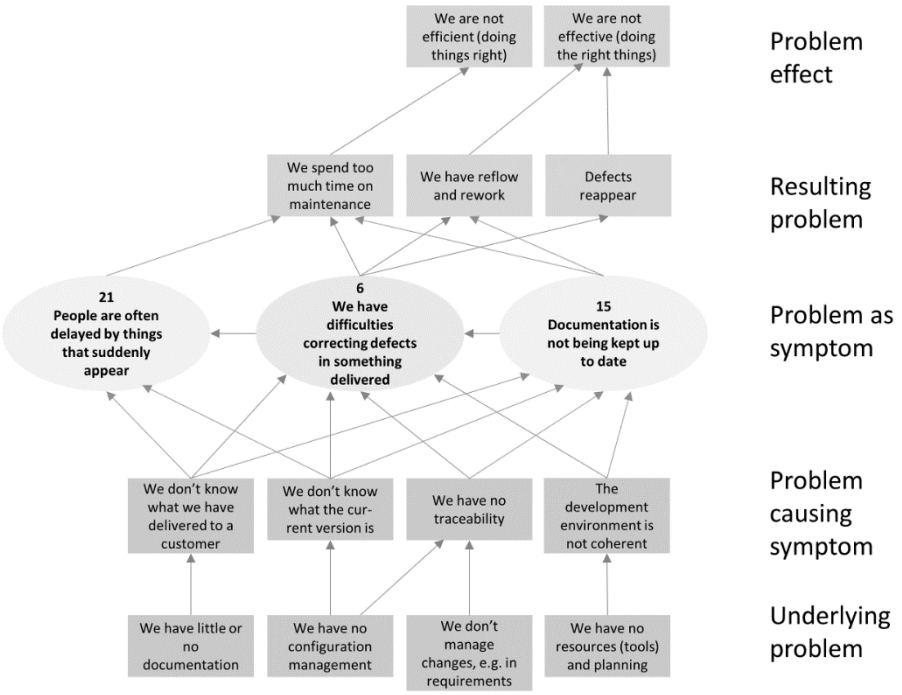

Fig. 2. A cognitive map of symptom no. 6 and the relationships to symptom no. 15 and symptom no. 21.

Thus, Figure 2 shows that symptom no. 15 'Documentation is not being kept up to date' and symptom no. 21 'People are often delayed by things that pop up / are not planned for' are closely related to symptom no. 6.

Another aspect we identified was that there were problems at different levels. We have mapped things at the most concrete level – the problem instantiations. However, there is also an effect of the whole map or network of problems. For example, the two effects – not doing the right things and not doing things right – together will cause the business as a whole to be bad. And the four problems that together are causing the three symptoms in Figure 2 will altogether form an unsound foundation for work.

Let us take another example. Let us look at symptom no. 15 (see Figure 3).

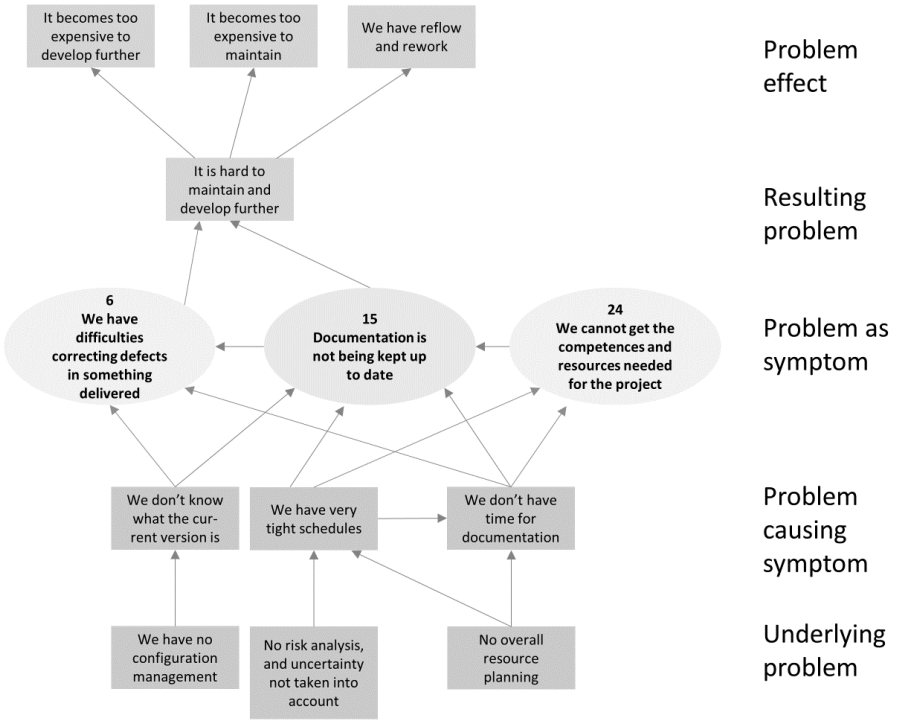

Fig. 3. Example of a cognitive map – here, of symptom no. 15 – and how it is related to symptom no. 6 and symptom no. 24

For the 15th symptom, 'Documentation is not being kept up to date' in Figure 3, we find again that the symptom leads to rework – but starting from different underlying problems. Again, looking at the meta-level or the 'whole', we can conclude that the overall effect is that it will lead to bad business and that the level causing the symptom(s) as a whole will make it nearly impossible to maintain the product.

Another interesting observation in Figure 3 is that a problem can cause another problem directly or through another – more indirectly, so to speak. An example is 'We have very tight schedules.' That can lead directly to symptom no. 15 but it can also lead to 'We don't have time for documentation' that can in turn lead to symptom no. 15 'Documentation is not being kept up to date'.

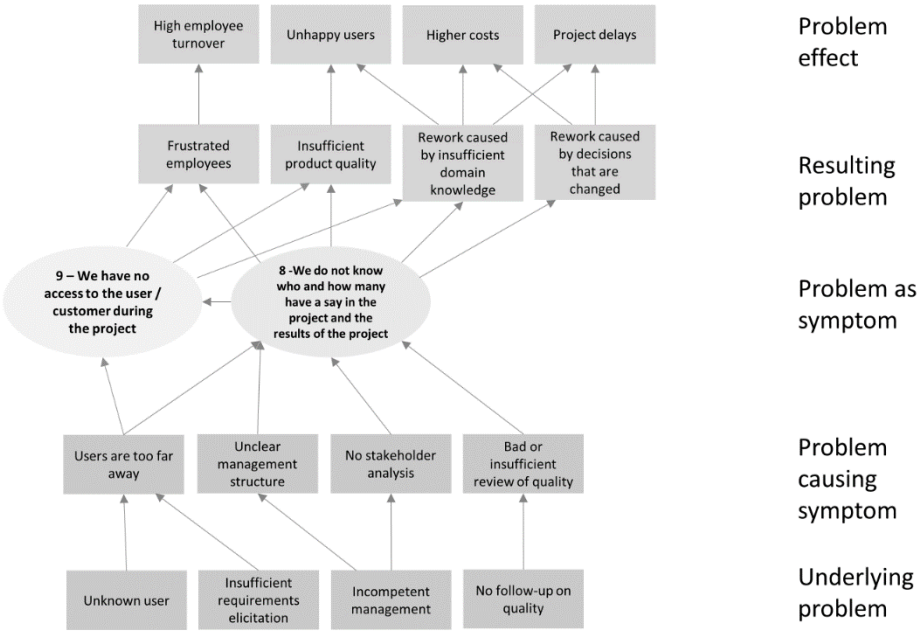

Fig. 4. An example of a cognitive map – here, symptom no. 8

In the next example shown in Figure 4 it is even more obvious that the higher-level problems can be the same as some of the lower-level underlying problems.

Furthermore, for symptom no. 8 there is again a relationship to another other symptom, namely no. 9 'We have no access to the user / customer during the project'. So let us now take a closer look at symptom no. 9 (see Figure 5).

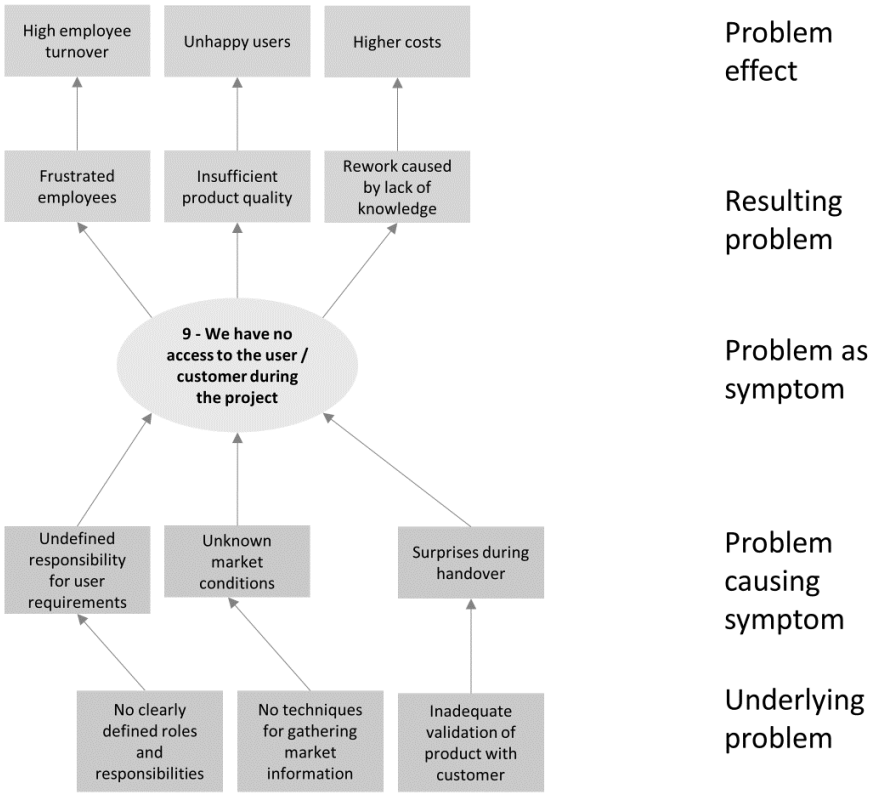

Fig. 5. An example of a cognitive map – here, symptom no. 9

When we compare Figure 4 and Figure 5, we find that the layers above the core symptoms in the middle are nearly the same for symptom no. 8 and symptom no. 9, whereas the layers below are quite different for the two symptoms. This means that the root causes of symptom no. 8 and symptom no. 9 are very different. It also means that the solutions – or practices – that we need to apply to deal with these two symptoms are quite different.

As our last part of working with the maps we compared the root causes in our 32 maps with the practices of the well-known CMMI maturity model [1, 2]. However, to do that we first converted the cognitive maps of the 'problem as difficulties' into 'problem as solutions'; a process described by John Venable [31]. E.g. a symptom such as "hard to plan testing" will be converted to "easy to plan testing".

Specifically we (1) Reversed all nodes in the map to make the undesirable desirable and desirable nodes undesirable; (2) Added or modified text for poles of nodes so it matched the opposite pole; (3) Made all nodes begin with a verb in the imperative

(command) tense followed by an object noun. E.g. ‘eliminate or reduce causes’, ‘solve or alleviate problems’, ‘improve symptoms or implications’ and so on.

Finally, after the conversion of the 32 cognitive maps of inter-related problems we ended up having 32 maps of inter-related solutions. We then compared the roots of the maps to the widespread maturity model, CMMI. To our surprise the roots in the maps corresponded very clearly to practices described and recommended in CMMI. E.g. for symptom no. 8, the CMMI model would suggest improving:

- Requirements Development (RD) – Develop Customer Requirements, Elicit Needs to address the unknown users and insufficient stakeholder analysis.
- The entire Process and Product Quality Assurance (PPQA), for the general lack of quality focus, and the Generic Goals to address the missing institutionalization and incompetent management.

And, for comparison, symptom no. 9 would suggest improving:

- Again, RD – Develop Customer Requirements, Elicit Needs to address the lack of techniques for gathering market information, but also the Generic Practice 2.4: Assign Responsibility to address the lack of defined roles and responsibilities.
- Validation (VAL) to establish the adequate validation practices.

So to conclude this mapping section of the paper, we have found that it was very useful to use cognitive maps to obtain an overview of the symptoms and the relationships either directly or through other symptoms.

6 Evaluation

An important part of using DSR as the research method is that you have to evaluate the design [35]. Typically, you start by having one or more formative evaluations where the result is used to ‘form’ the artefact. You end up with a summative evaluation that ‘sums up’ or concludes your research. At the beginning, the first formative evaluations may be in artificial settings but later you aim to have real users in the real context with the real problem – real, real, real – a so-called naturalistic evaluation [35].

We have used a framework for evaluation called FEDS by John Venable et al. [35] to plan and carry out our evaluation. For our first formative evaluation we developed a tool that we made available on a website. The tool presents a person in an organization for each of the 32 symptoms to ‘score’ it, seen from the person’s perspective.

The 32 symptoms are grouped in 5 categories:

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

These categories were chosen mainly because of the establishment of a focus while the questionnaire is completed – instead of only having a long list of questions. The categories were identified through discussions between the assessors. Examples of the arguments:

- We need a category for being in **control** during a project, because it is an inherent part of being mature.
- Typical problems we often discuss with team members are problems with requirements and the agreement on **what to deliver**.
- Another important aspect of maturity models are personal and organizational **capabilities**.
- If the organization around a project should be able to support the project, the necessity of project insight and **overview** is obvious.
- And finally, **quality** is what it is all about behind the maturity models.

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, etc.)					
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 task					
22	We do not know our performance on the different types of tasks					
23	We do not utilize resources effectively across projects/teams					
24	Employees shifts between several task/projects through the week					
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					

Version 1, © Whitebox 2019

Fig. 6. Here are the 32 symptoms shown in the statement questionnaire that we actually used

We asked nine people from different development companies to 'score' all 32 symptoms and give their opinion about how easy it was to understand the symptom statements. The nine people were one CEO, one R&D leader in change, five middle managers, two project managers, all with strong interest in change.

After having evaluated the website with nine users, we analysed the outcome and changed some of our statements and the built-in relationship between problems. The tool was updated so that an e-mail with the recommendations would be returned automatically once 'scoring' had been completed. The website was then tested again, also with nine users. This time the focus was on analysing the algorithm in the tool in order to strengthen the recommendations. Again, this led to some small changes in the formulations of the statements.

Finally, as our summative evaluation we decided to evaluate the website tool against a classic CMMI [1] assessment of maturity. The two assessors filled in the questionnaire separately following a CMMI assessment to check if they had the same answers to the questions.

Same score	48%
1 score-step in difference	37%
2 score-steps in difference	16%
3 score-steps in difference	0%
4 score-steps in difference	0%

Table 1: Differences in the score of one assessed company
(e.g., a 2 score-step in difference could be neither nor instead of Totally agree)

During the exercise and following discussions and analysis, it became clear that it is not possible to derive a maturity level for an organization from the symptoms alone; we can only pinpoint the main weaknesses that it is important to address.

We started to identify which overall problem-related capabilities the symptoms reflected in an organization showing the ability to run projects successfully and ended up with ten important themes: (1) Project management; (2) Control across the organization; (3) Proactive management; (4) Clear project goals; (5) Capable development organization; (6) Quality of results and products; (7) Control of work products and product parts; (8) Management insight and involvement; (9) Tools and support; (10) Continuous improvement and learning. The algorithm was adjusted to calculate a score for the 10 themes as a basis for recommendations on what is most important to focus on and to improve.

There are obvious links between the ten defined main themes related to the overall ability to successfully run projects and the processes. This link can be used to pinpoint the recommendations but not the maturity level. The reason is that the themes are each based on several processes, but it seems to be clear which themes are aggregated from which symptoms. This exercise was used to strengthen the questions, as well as the 10 themes. E.g., we did several analyses to evaluate the mapping of the practices in the CMMI processes (maturity level 2 and 3) to the 10 themes, which ended in the result presented in figure 7.

PP, PMC	**Project management**
IPM, SAM	**Control across the organization**
RSKM	**Proactive management**
RD, REQM	**Clear goals for the projects**
TS, PI	**Capable development organisation**
VER, VAL	**Quality of results and products**
CM & REQM	**Control of work products and product parts**
PPQA, MA, DAR	**Management insight and involvement**
OPD	**Tools and support**
OPF, OT	**Continuous improvement and learning**

Fig. 7. Mapping of CMMI processes to the 10 themes

Figure 7 shows the scoring of the themes presented in one of the company trials. In the left column you can see that 'Control of work products and product parts' and 'Project management' score highest, meaning that the company in the example has good control of these two themes. Likewise, 'Continuous improvement and learning' and 'Management insight and involvement' are the two lowest scoring themes, meaning that this is where the company in the example must focus and improve.

Rank	Score	Ability to run projects successfully	No.
2	**43%**	**Project management**	1
8	**22%**	**Control across the organization**	2
4	**31%**	**Proactive management**	3
6	**28%**	**Clear goals for the projects**	4
5	**30%**	**Capable development organisation**	5
3	**33%**	**Quality of results and products**	6
1	**47%**	**Control of work products and product parts**	7
10	**14%**	**Management insight and involvement**	8
7	**25%**	**Tools and support**	9
9	**21%**	**Continuous improvement and learning**	10

Fig. 7. Example of a result scoring the symptoms for an organization

We also started to discuss a mapping between the lowest levels of problems and the specific practices in the CMMI model. We will continue this work because we believe we can find connections between the 10 themes and specific practices at the processes in CMMI. We also believe this work will strengthen the symptom-based model, for example if we find some practices in CMMI not addressed at the lowest level of problems, then an important symptom may be missing.

7 Results and discussion

We have two sets of findings to report. First, we will validate the use of cognitive maps. Based on the links from symptoms to problems below and above, we confirmed that some symptoms where related, because the symptoms linked to the same problems/causes. It proved to us that the development of cognitive maps gave an insight into how symptoms are related seen from a problem/cause point of view. We even identified clusters of problems, which made good sense, and gained an understanding of the symptoms.

For example, looking at symptom no. 6 in Figure 1, it is strongly related to symptom no. 15, as both symptoms share many problems and causes. Symptom no. 21 is seen to be slightly more weakly coupled. As a health check we formulated the relationship between problems, causes and symptoms as:

'If you do not have a good foundation for work (the problem level just below the symptoms), then it is very difficult to keep the documentation up to date. And if you do not have updated documentation, it is difficult to correct something that has been delivered. If maintenance is difficult, it typically generates ad hoc rework, which annoys people.'

To take another example, in Figure 3, symptom no. 8 strongly links to both symptom no. 9 and no. 11 at the upper problem level. Health check: does it make sense to explain the relationship?

'*If we do not know who decides what in the project, it is difficult to make clear decisions and this will typically also include a lack of access to users during the project (since we do not know who decides). All three symptoms lead to lack of motivation, initiated by frustration, failure to fulfil quality and a lot of rework.'*

We find that cognitive maps help a lot to clarify and structure explanations of the main reasons for major problems. They also help to qualify and strengthen the symptoms, as well as the model. We believe that, over time, working with the model will identify 'weak' symptoms, which then will be updated.

The second finding came from our summative evaluation against an organization where two of the authors had performed a CMMI assessment. As expected, it was clear that, from the symptoms, it is not possible to derive a maturity level for an organization – we can only pinpoint the main weaknesses that it is important to address. The reason for this lies in the basic elicitation process in the assessment. In a traditional assessment, the assessors interview the participants following a defined set of process areas from a model, that is, covering all needed practices in the domain. This way, practices in which the interviewees are unconsciously incompetent are also covered, and this may very well uncover weaknesses in capabilities. A capability weakness is a risk that must be identified in an assessment, for compliance reasons, and whose mitigation must be prioritized for process improvement reasons. When we ask specifically about symptoms, we will get answers about to the largest experienced problems. We will not hear about the potential problems, the risks, that the classical assessment approach would uncover. We would also only hear what the interviewee finds is problematic and not what a representative group of participants finds is problematic. That is why the symptom-based

approach will never be able to provide a compliance result (maturity level), but is expected to be reasonably valid to produce focused improvement recommendations and make these recommendations after significantly less effort. However, we find there is a link between the 10 themes and the processes in CMMI. We will continue that research.

On the whole, our approach of looking at symptoms is somewhat similar to the SPINACH method [36] developed by Information Promotion Agency in Japan. It has a checklist of about 150 potential symptoms that could trigger further analysis to produce an affinity diagram of a causal system.

We also tried to 'switch over' a map to the opposite, following Venable [31]. In Figure 8 we have switched around the map that was shown in Figure 5.

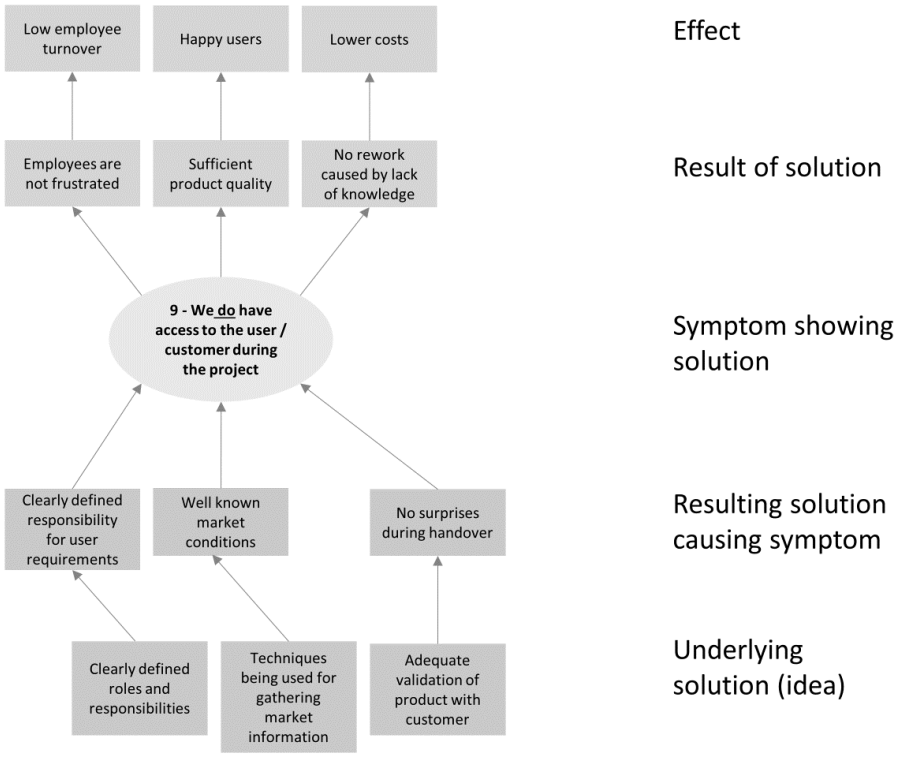

Fig. 8. An example of a design solution map – here, symptom no. 9 from Fig. 5 has been switched around.

In Figure 8 we can then see that instead of having root causes at the bottom, we have potential core solutions such as 'clearly defined roles and responsibilities', 'techniques

being used for gathering market information' and 'adequate validation of product with customer'. Interestingly enough, these core design solutions are very similar to practices in CMMI [1] such as Project Planning Specific Practice 2.4 Plan the Projects Resources, Requirement Development Specific Practice 1.2 Elicit Needs, and Validation Specific Practice 2.1 Perform Validation. Therefore, there is a clear connection to CMMI at practice level.

8 Conclusion

We have answered our research question, how can we design a simple and fast approach to establish recommendations for an organisation?, by using a mapping of symptomatic problems, causes and effects, and the relationships identified between them, to design a tool that can help determine what the most urgent improvement areas are in a given company and come up with recommendations.

We have presented a sub-set of the 32 cognitive maps we have created, one for each of the symptoms from which we started out. We have shown how causes, underlying problems and effects can be related and apply across symptoms. We have also found that a group of problems or effects seen as a 'whole' can lead to bad business, reduced employee motivation or other major meta-level effects for a company.

To evaluate our 32 cognitive maps, we have built a tool where people can 'score' all 32 symptoms and obtain overall recommendations, but never a maturity level or other compliance results. We will ask users about their experiences with the tool. We will use these data to strengthen the model and the recommendations.

We have planned and carried out a trial with a company, where we asked a manager and a developer to try out the tool. After that, for each symptom they indicated as 'totally agree' we have asked them to identify the most relevant problem-causing symptom, underlying problem, as well as resulting problem and the problem effect. Thereby we have identified a cause-and-effect link from underlying problem to the effect orf the problem. This can then be used to identify the best recommendation in relation to improving practice. And – if they follow the recommendation - hopefully reducing or even eliminating the symptom.

The threats to validity of our findings are multi-faceted. First, the 32 symptoms derived from 600+ assessments may not be representative of all assessments. Second, our cognitive mapping – again based on our assessment data – of cause and effect may leave out things that are important in other companies. Third, we have used to CMMI to identify recommendations which could introduce some bias. And fourth, the use of our tool may be prone to mis-communication i.e. users not understanding the questions and how they are formulated.

Based on our research in this paper, we have identified at least two interesting topics for further research.

The first thing is to use a tool, which enables the possibility of combining all the cognitive maps into one showing all symptoms and different types of problems and causes. We believe this map brings new knowledge on how symptoms are related and how problems and causes are related. It will offer new possibilities for improving the

model and strengthening the symptoms foundation in problems and causes. We will continue this work in the future.

The second thing is to continue the investigation into how the 10 themes are related to the processes in CMMI. We saw that the lowest level of problems in the cognitive maps for each symptom has a relationship with the specific practices in CMMI processes. For example, in Figure 3 the underlying problem 'No overall resource planning' is caused by not performing the process Project Planning Specific Practice 2.4 Plan the project's resources. We will map these relationships and see how the themes and symptoms are related to CMMI and use that knowledge to see if some CMMI practices are missed, which could indicate a missed problem, cause or symptom. With these relationships in place, we can guide a distressed manager towards the specific practices where improvements are highly likely to be the most beneficial.

References

1. Team, C.P., *CMMI for Development, Version 1.3, Pittsburgh, 2010.* Software Engineering Institute, Carnegie Melon University, 2018. **2**.
2. Team, C.P., *CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, Version 1.1, Continuous Representation.* CMU/SEI, 2002.
3. 33060, I., *ISO 33060 - Information technology — Process assessment — Process assessment model for system life cycle processes*. 2020, International Standardisation Organisation. p. 66.
4. Gibson, D.L., D.R. Goldenson, and K. Kost, *Performance results of CMMI-based process improvement*. 2006, Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst.
5. Goldenson, D. and D.L. Gibson, *Demonstrating the impact and benefits of CMMI: an update and preliminary results.* 2003.
6. McCurley, J. and D.R. Goldenson, *Performance effects of measurement and analysis: Perspectives from CMMI high maturity organizations and appraisers*. 2010, ACCEL SOFTWARE ENGINEERING PITTSBURGH PA.
7. Falcini, F., G. Lami, and A.M. Costanza, *Deep learning in automotive software.* IEEE Software, 2017. **34**(3): p. 56-63.
8. Johansen, J. and M.N. Andersen, *ProductAbility – Evne til produktudvikling ("The product development ability")*. 2014, Hørsholm, Denmark: DELTA.
9. Miles, M.B., A.M. Huberman, and J. Saldaña, *Qualitative data analysis: A methods sourcebook*. 2018: Sage publications.
10. Liker, J., *The toyota way*. 2004: Esensi.
11. Ohno, T. *"Ask 'why' five times about every matter."*. 2006 [cited 2021 30 June]; Available from: https://www.toyota-myanmar.com/about-toyota/toyota-traditions/quality/ask-why-five-times-about-every-matter.
12. Lee, D., J.W. Gu, and H.W. Jung, *Process maturity models: Classification by application sectors and validities studies.* Journal of Software: Evolution and Process, 2019. **31**(4): p. e2161.

13. Rout, T.P., *ISO/IEC 15504—Evolution to an international standard.* Software Process: Improvement and Practice, 2003. **8**(1): p. 27-40.
14. Hoermann, K., et al., *Automotive spice in practice.* Rocky Nook, 2008: p. 75.
15. Kroenke, D., *Using MIS 2013*. 2013: Pearson Education UK.
16. Dumdum Jr, U.R.B., *An approach to problem formulation in ill-structured situations in information systems development.* 1993.
17. Checkland, P., *Soft systems methodology.* Human systems management, 1989. **8**(4): p. 273-289.
18. Checkland, P. and J. Poulter, *Learning for action: a short definitive account of soft systems methodology and its use for practitioner, teachers, and students*. Vol. 26. 2006: Wiley Chichester.
19. Checkland, P., *Soft systems methodology: a thirty year retrospective.* Systems research and behavioral science, 2000. **17**(S1): p. S11-S58.
20. Rittel, H.W. and M.M. Webber, *2.3 planning problems are wicked.* Polity, 1973. **4**(155): p. e169.
21. Buchanan, R., *Wicked problems in design thinking.* Design issues, 1992. **8**(2): p. 5-21.
22. Alford, J. and B.W. Head, *Wicked and less wicked problems: a typology and a contingency framework.* Policy and Society, 2017.
23. Snowden, D.J. and M.E. Boone, *A leader's framework for decision making.* Harvard business review, 2007. **85**(11): p. 68.
24. Snowden, D., *Naturalizing sensemaking.* Informed by knowledge: Expert performance in complex situations, 2010: p. 223-234.
25. Eden, C., *Cognitive mapping.* European Journal of Operational Research, 1988. **36**(1): p. 1-13.
26. Eden, C., *On the nature of cognitive maps.* Journal of management studies, 1992. **29**(3): p. 261-265.
27. Eden, C., *Analyzing cognitive maps to help structure issues or problems.* European Journal of Operational Research, 2004. **159**(3): p. 673-686.
28. Williams, T., et al., *The effects of design changes and delays on project costs.* Journal of the Operational Research Society, 1995. **46**(7): p. 809-818.
29. Eden, C. and F. Ackerman, *Making Strategy. The Journey of Strategic Management*. 1998, Londres: Sage Publications.
30. Ackerman, F., C. Eden, and S. Cropper, *Getting started with cognitive mapping: tutorial notes.* Strathclyde: Strategic Decision Support Research Unit, Strathclyde University. Retrieved November, 1992. **18**: p. 2008.
31. Venable, J.R. *Using Coloured Cognitive Mapping (CCM) for design science research*. in *International Conference on Design Science Research in Information Systems*. 2014. Springer.
32. Hevner, A.R., *A three cycle view of design science research.* Scandinavian journal of information systems, 2007. **19**(2): p. 4.
33. Walls, J.G., G.R. Widmeyer, and O.A. El Sawy, *Building an information system design theory for vigilant EIS.* Information systems research, 1992. **3**(1): p. 36-59.
34. Pries-Heje, J. and J. Johansen, *Spi manifesto.* European system & software process improvement and innovation, 2010.

35. Venable, J., J. Pries-Heje, and R. Baskerville, *FEDS: a framework for evaluation in design science research.* European journal of information systems, 2016. **25**(1): p. 77-89.
36. Norimatsu, S., T. Kishi, and N. Wada. *Development of "SPI Strategy Framework" and Its Application*. in *European Conference on Software Process Improvement*. 2018. Springer.