Project Smells for Early Detection of Problems with Benefits Realization

Sinan Sigurd Tanilkan
Department of Applied Research in Information Technology
Norwegian Computing Center
Pb. 114 Blindern, Oslo, Norway
Center for Effective Digitalization of the Public Sector
Simula Metropolitan
Pb 4, St.Olavs Plass, Oslo, Norway
Email: sinan@nr.no
0000-0003-4216-5172

Jo Erskine Hannay
Center for Effective Digitalization of the Public Sector
Simula Metropolitan
Pb 4, St.Olavs Plass, Oslo, Norway
Email: johannay@simula.no
0000-0002-8657-7593

***Abstract*—Although substantial research has provided guidance on how to identify and manage the benefits of new software solutions, ensuring the realization of those benefits remains a challenge. Inspired by the notion of code smells for software quality, we develop a concept of *project smells for benefits realization*. We conducted 22 in-depth interviews with participants in nine public-sector digitalization projects, and elicited seven project smells: 1. Dilemma between enthusiasm and formality, 2. Situational differences, 3. Resistance to realization, 4. Slipping opportunities, 5. Loss of focus due to project size, 6. Lacking commitment, 7 Insufficient contact with recipients. We argue that these project smells are a complement to traditional project metrics which focus on time, cost and scope, or the evaluation of benefits after a project is finished. Each smell comes with a set of questions intended to help practitioners identify the odour of their projects. The intention is that project smells can function as low-cost, early indicators helping practitioners adjust work readily and rapidly to ensure benefits realization of their software development investments, thereby focusing actively on the project's product, rather than myopically on the project itself.**

Keywords*— *Software Project; Continuous Product Development; Benefits Realization; Agile; DevOps; BizDev.

I. Introduction

Keeping track of the status of a software engineering initiative is important for all who have stakes in investing, developing and benefiting from the system under development. The purpose of such monitoring is to ensure project and product success.

The iron-triangle metrics time, cost and scope are often habitually monitored during software development. However, keeping track of the benefit of the system under development is not regularly done [1]. This may result in suboptimal choices in terms of what functionality to develop early and may lead to poor decisions on development progress, and even termination, based on time, cost and scope alone. Indeed, a recent study found that software professionals perceive that important decisions in software development are mostly based on rationales in terms of time, cost and scope, rather than on benefit or the benefit/cost ratio, but that they think decisions should be based on benefit or the benefit/cost ratio to a greater extent [2].

Early on, Baccarini made the distinction between project management success and product success and defined overall project success as a healthy balance between the two [3]. Project management success is delivering on time, cost and scope, while product success is delivering software that generates value, or benefit. In other words, software engineering is, in many cases, suffering from myopia on project management success, even though practitioners, policy makers and academics call for a greater focus on organizing work to ensure that the software is capable of delivering benefit.

The field of *benefits management* arose to address the lack of attention on benefits in development initiatives [4]–[10]. Even though studies suggest that projects that engage in benefits management activities during project execution [1] are more successful on delivering benefit (but also on time, cost and scope), the adoption of such activities has been low [1], [11]. Of several reasons for this, we highlight that benefits management is for the most part formulated at the portfolio and project program level (such as in Managing Successful Programs®), and benefits management activities during development mostly relate to strategic and organizational aspects, such as paying attention to the business case and the benefits realization plan, but with no operational means for doing so.

Considerations of the benefit of a system under development may belong at the strategic level of business cases and at the organizational level for planning how to use a system under development to produce value in organizations and in society. Nevertheless, it has been argued convincingly that to realize such high-level plans, one must also move benefits considerations from the strategic level onto the operational level. A step on this route is to measure the realization of intended benefits [4], [5], [10]. If this is done in an incremental-development setting, measurements on how beneficial an increment is can be fed back to provide project learning on benefit for successive increments. However, such data is usually not used in this way, but rather collected after a project is completed as part of reporting [5], [10].

Methods have been developed for monitoring how well the project is doing in producing valuable software during project execution, for example, using *lead indicators* of benefits [5], [12]. Another approach is to use *benefits points* (and related

techniques) [13]–[16] to estimate and keep track of a system's potential for realizing benefits during project execution [17]. This includes using lag indicators to update lead indicators to take advantage of learning during project execution. However, getting industry to take such techniques into use in a broad scale requires integrated tools for managing daily work (e.g., Jira), which are under development but not operational [18], and an appropriate organizational mindset, which seems to depend more on vogue grey literature than academic studies.

In this article, we approach the problem of keeping track of software engineering work, and specifically with respect to benefits creation, not from a managerial standpoint, but from the point of view of the working software engineer. To update lead and lag indicators for benefit one needs ground truth from those close to development and close to stakeholders affected by the system. To provide a lower threshold utility, and inspired by the work of Fowler et al. on code smells [19], we elicit and elaborate the notion of *project smells for benefits realization*. While *code smells* help to identify software design flaws, these *project smells* help to identify concerns regarding the benefits from software projects. We will argue that project smells can help provide a shared view of project status between software engineering teams and their managers. This addresses the issue that project participants may be aware of projects that are in trouble, while management is unaware [20]. This topic has been studied under the term employee silence – when individuals remains silent about project concerns or problems, resulting in a situation where the project's true status is not known [21]. Project smells for benefits realization can help give voice to such concerns.

Section II presents relevant background. Section III describes the development of the project smells notion, moving from interviews, through coding and concept development to concept operationalization in smells. We present and discuss the results in Sections IV and V. We consider limitations in Section VI and conclude in Section VII.

II. Background and Previous Work

The term *project smells* has been used previously, with different meanings. The first academic mention of project smells in 2007 focused on end-of-project retrospectives. No elaborations of the project smells were provided, but the proposed purpose of project smells is to "... alert us to a broken or ill-fitting process" [22].

Three months later, the term project smells was used again, in the context of software testing. Three categories of test smells were proposed: 1) test code smells, 2) automated test behaviour smells, and 3) project smells [23]. Factors to identify project smells in this context are: i) buggy tests, ii) developers not writing tests, iii) high costs of test maintenance, and iv) bugs in production. The purpose of project smells in this context is that "... project smells are likely to be the first hint they get that something may be less than perfect in test automation land" [23].

The third academic reference for project smells was published recently, within machine-learning project management [24], where project smells are presented as a holistic view of software quality in machine learning projects, and code smells are considered to be a part of project smells. Here, characteristics of project smells are lack of 1) dependency management, 2) version control, 3) unit testing, 4) proper configuration of continuous integration, and 5) effective static analysis tooling [24]. Project smells are evaluated using a static analysis tool on the source code, the data and the tools configuration in the projects. The purpose of project smells in this setting is thus to provide feedback to practitioners on their use of a set of practices for machine-learning quality assurance.

Smells have also been used for the assessment of agile practices in organizations. Agile smells [25] constitute a catalogue of practices that are considered suboptimal in an agile context. Simlilarly, Mike Cohn, has written an article about scrum smells [26] – a catalogue of suboptimal practices in scrum.

The above smells revolve around appropriate work practices and delivering on time, cost and scope, and also software (intrinsic) quality. What is absent, then, from our vantage point, are smells concerning the benefits of the software.

The field of managerial problem solving brings relevant perspectives to the table and regards problem solving as consisting of two parts: 1. problem formulation and 2. problem solving [27]. Project smells for the early detection of problems (ground truth) with benefits realization falls into a subset of the first category, specifically *problem detection*, concerning observations "... that events are taking an unacceptable trajectory and may require action" [28]. Once a problem is detected, people can choose from a set of option categories: look for more information, pay more attention to related events, attempt to identify the underlying problem, discuss the concern with others, explain away the observation or take action to manage the problem by mitigating actions or accepting a change in situation and updating plans and goals [28].

However, the above options are only available once the problem has been identified. An interesting aspect of managerial problem detection is its contrast to the detection of managerial opportunities and crises in terms of the relevant stimuli. While both opportunities and crises are often stimulated by one single idea or triggering event, problems often require multiple stimuli, and problem-stimuli are often milder than stimuli from opportunities and crises [29]. Also, decision makers have a tendency to desire more information about problems before they act [30]. This makes problem detection more complex and less clearly delineated than other critical events in management.

Then, the three most important factors for identifying a problem is: *expertise*, *stance* and *attention management* [28]. While expertise is intuitively understood, and most would agree that an experienced project manager is more likely to spot project problems, stance and attention management warrants a short description. Stance is a person's position towards a situation [31]. Stance (or general alertness) can range from denial (nothing can go wrong) to being confident that any

obstacle can be overcome, to an alertness that problems may arise, to hysteria (over-reacting to every minor indication) [28]. Attention management, on the other hand, focuses on what is monitored (and ignored) [28]. This is central to our discussion, because we have a tendency to overlook relevant information, even when it is just in front of us [32], [33]. Although the project smells are unlikely to affect people's expertise and stance, it is our hope that the project smells can help guide practitioners' attention to factors that help them with early detection of problems with benefits realization.

III. Research Method

The present study is an elaboration of a particular concept that was elicited in a larger qualitative study based on thematic analysis of interview data. Several concepts emerged in that larger study, and it is the further elaboration of the concept *Characteristics of projects that affects the realization of benefits* that is presented here. More information about the full study can be found at [34]. Focusing on a single concept from a larger study is recommended to allow one to go into detail in that particular concept [35]. That concept is the basis for the seven project smells for benefits realization that will be presented in Section IV.

The research was conducted using the Stepwise-Deductive Induction (SDI) method [36]. The SDI method is a structured qualitative method for building concepts and theories, grounded in empirical data. Our intention is not to develop new theories, so the last step of the SDI method – theory development – is omitted. The phases of the SDI method used here are:

1) Case selection and data generation
2) Processing of raw data
3) Coding
4) Code grouping
5) Concept development

Moving from one phase to the next is an inductive move. Deduction is used to test the results from each phase by comparing the results to the data that formed the input to the phase. Abduction is used in the latter phases (primarily during concept development), to find plausible explanations for the observations [37].

In addition to the phases of the SDI method we added a sixth phase, which operationalizes the developed concepts into actionable tools; in this case, project smells and questions to keep in mind to become aware of the smells:

6) Operationalization of the concept

Braun & Clarke [38] distinguish between different approaches to thematic analysis. The *neopositivst* approach, at one end of the scale, focuses on objective and unbiased coding. In that approach, it is common to use a predefined codebook and have multiple coders, so that agreement between coders can be measured numerically as a measure of coding reliability [39]. At the other end of the scale is the *reflexive* approach, where coding is "... open and organic ..." [38], with no predefined codebook. An important distinction between the two ends of the scale is when themes or concepts are developed. In the neopositivst approach, themes are developed early, often prior to coding, while in the reflexive approach, themes and concepts are the final outcome of the analysis. The SDI method used here is at the reflexive end of the scale, where codes are developed inductively from the data. Still, coding reliability, which is often not a concern in reflexive thematic analysis, has a strong focus in the SDI method. While the neopositivist thematic analysis approach to coding reliability is often handled by codebook design and using multiple coders, coding reliability in the SDI method is handled by adhering to strict coding rules (see Section III-C).

Coding for this study was conducted by the first author only. To ensure conceptual clarity at the higher levels (Steps 5 and 6), the code groups of the concept under study and their operationalization in smells were discussed extensively between both authors.

The following sections describe how the the six phases were applied in this study.

A. Case Selection and Data Generation

Due to the low adoption of benefits management in practice [40], it is challenging to find organizations where its use can be studied. The Norwegian Digitalization Agency has a funding program for public-sector digitalization projects, where one of the conditions for funding is the active use of certain benefits management practices, as described in [41].

We invited all the projects that received funding in 2016 to participate in the study. Nine out of twelve projects chose to participate. All included projects had a duration of three years, except one, which had a duration of two years, and funding was granted up to 50% of the net project costs with an upper bound to funding at NOK 15 million (approx. USD 1.9 million at project completion time).

All the studied projects involved the creation of a new software solution for digitalization in the public sector. The new software included solutions for:

- data sharing,
- unique data storage (to avoid a situation where data is duplicated and out of sync across organizations),
- providing individuals and organizations with self-service to public-sector data and applications,
- automating previously manual and/or paper-based case processing,
- guiding individuals and organizations on how to use public-sector services.

Examples of benefits from the new software solutions included more efficient use of resources in the public sector, improved quality of data in the public sector, faster response times when interacting with the public sector and improved rules of law.

Data was collected through 22 face-to-face interviews with professionals involved in the projects. The interview questions are available at [34]. We chose a semi-structured approach in order to follow (other) topics that respondents brought up as

relevant to benefits management and realization. Interview participants included project sponsors, project managers, people responsible for benefits realization, project team members and one benefits recipient.

Interview duration ranged from 25 to 120 minutes, depending on the amount of relevant information the respondent had to provide. For all interviews, two researchers and one respondent took part, except for five interviews. Four interviews were conducted with one researcher and one respondent, and one interview was conducted with one researcher and two respondents. Interviews were conducted face-to-face, in the premises of the studied organizations.

All interviews were recorded using an audio recorder and notes were taken. Due to strict confidentiality agreements, none of the collected raw data is made available; neither are organization names nor exact project topics.

B. Processing of Raw Data

The recorded audio files were transcribed, resulting in 612 pages of transcribed text. After transcription was completed all audio files were listened to while simultaneously reading the transcribed texts to ensure correct transcriptions.

C. Coding

All the transcribed texts were coded using NVivo (release 1.7.1). In total, 274 codes related to the concept focused on in this study were created.

A two-step code test was applied to all codes to ensure that the codes represented the respondents' statements [36]. This approach is designed to reduce the potential biases of having only one coder [39], when relevant. The applied code tests suggested by [36] emphasize groundedness in data, and semantic codes, rather than mere sorting codes: 1. If the code could have been created prior to seeing the data, this is considered an *a priori* code, and a different code should be created based on the data. 2. If the code only labels topics in the data, e.g. "quantification of benefits", this is considered an unnecessary sorting code, and a different code should be created that reflects what the respondent expressed, e.g., "Numbers makes people lose interest".

To give the reader an impression of the codes and the respondents' statements that the codes represent, examples of codes and corresponding extracts from the interviews are included in the results section (Section IV).

D. Code Grouping

Codes were grouped thematically, using a code grouping test. Rather than applying this test at the end of the phase (as suggested by [36]), the grouping test was used as a guide or condition when placing codes into groups. The grouping test checks that when adding a code to a group, the group should still be thematically different from the other groups and the content of the group should still be consistent. If a code cannot be placed in any group (and still fulfil these conditions), a new group should be created.

Due to the large number, the code groups were divided into two levels. This approach is supported by [36, p. 210], who suggests that when there are more than 3–5 code groups, it can be useful to organize them into more than one level. The five high-level and eight low-level code groups are the primary building blocks of the concept under elaboration here, and the results section (Section IV) is organized accordingly.

E. Concept Development

The concept *Characteristics of projects that affects the realization of benefits* evolved using abduction – moving back and forth between the code groups and relevant background knowledge and theories [42] to consolidate the concept. Recall that the abduction process was performed in the context of the larger study, and the concept was generated in relation to the other concepts in that encompassing study. All these were tested by considering how well they described different subsets of code groups.

F. Operationalization of the Concept

From that concept we distilled the project smells for benefits realization. Within each high-level code group, this was done by operationalizing the lower-level code groups into actionable tools for practitioners; namely indicators (smells) and actions (questions to be asked). These proposed indicators and actions can be found as subsections of Section IV, starting with "Smell:".

We tested the notion project smells with other candidate denotations, such as "characteristics of benefits", "project heuristics", "product smells" and "project status detectors" by presenting them to practitioners with experience from software projects, with continuous product development [43], [44], and with project managers of construction projects, to get feedback and see how well the different denotation resonated in the different settings. The denotation "project smells for benefits realization" was kept.

IV. Results

The total SDI-analysis resulted in several conceptual topics, where one of these concepts is *Characteristics of projects that affects the realization of benefits*, which is the topic of this paper. The concept is built on the following high-level code groups (1–5) and low-level code groups (a–d):

1) Motivation
   a) The importance of caring about benefits
   b) Factors that affects peoples' motivation for benefits
2) Understanding
   a) Familiarity
   b) Proximity to domain
   c) The ability to understand resistance to benefits realization
   d) The ability to understand possibilities
3) Project size
4) Dependencies
   a) Changes in regulations
   b) Contributions from other organizations
5) The need and ability to reach benefits recipients

In the following subsections we describe the code groups, exemplifying them with extracts from the interviews and the accompanying codes. The project smells are given after the corresponding code groups have been presented.

A. Motivation

1) The importance of caring about benefits: Simply caring about the benefits of the system under development is reported to be important, both to motivate people to conduct measurements and to make the necessary adaptions when needed, as exemplified in the following excerpt (code: *Caring affects effort*):

> It is important that you care about realizing the benefits. If not, you will neither collect the necessary measurements, nor take the necessary actions when deviations from the plan occur.

While the above example illustrates that caring about the benefits is important for those working to provide them, caring is also important for those receiving benefits. After explaining an unanticipated benefit that was raised by an external organization, the following respondent stated that the benefits recipients' motivation was important in making this benefit materialize (code: *Interested recipients lead to unanticipated benefits*):

> They [the benefits recipients] are perhaps more than average interested ... and it turns out that they found benefits that we had not anticipated.

This observation is important, because it indicates that highly motivated recipients can lead to more, or further, benefits than anticipated.

2) Factors that affects peoples' motivation for benefits: Respondents report that people are easily motivated to work for society, as in the following excerpt (code: *Easy to motivate people for societal benefits*):

> It is not difficult to motivate [role] to work for society because they see that this is beneficial. They are driven by ... that is, they get energy from it. This is not merely an academic exercise, this is production of benefits for society.

In addition to the type of benefit, such as societal benefits in the above example, the way information about the benefits is shared also seems to affect peoples' enthusiasm for the benefits. In particular, verbalizing benefits, without emphasizing metrics and returns has been advocated (code: *Talk about benefits but not the numbers*):

> I think it helps just to talk about the benefits. Turn them into something concrete. That is, without focusing on the numbers, that is perhaps not something that motivates people.

Indeed, quantifying benefits seems to be a proper turn-off (code: *Numbers makes people lose interest*):

> Talking about the numbers in the excel sheet makes many people lose interest.

The above demonstrates a preference for verbal descriptions of the benefits over the numbers relating to benefits. This is further corroborated by the following statement highlighting story-telling (code: *Storytelling provided common direction*):

> I believe the pilot project was very important. But I was the only person who took part in that and the current project, and I think it was the story that was developed in the pilot project that people bought into. They immediately understood where we wanted to go, or the direction we were going in.

There also seems to be more engagement in benefits that people can envision thorough their daily work over the benefits reported in the benefits plans (code: *Driven by real, not artificial benefits*):

> *Respondent*: It has been a collaboration. And she has worked with them very much, also with the municipalities, in order to realize the benefits. But it's not... I don't think the benefits is what drives her. I think it is... That is, she genuinely cares about people, including the municipalities ... There are many qualitative benefits here that she is able to realize as a result of dialogue with the municipalities and other agencies.
>
> *Interviewer*: When you say that she was not driven by benefits, but that she genuinely cared. What is the difference between the two?
>
> *Respondent*: I think there is a difference between the benefits in the benefits plan and the benefits talked about in daily work, but I don't know how to explain it. One of them [the benefits] is perhaps something you have under your skin. Something you feel ownership of. You feel an ownership of a product, and by... You understand that this can lead to a process improvement for example, or that this simplifies things, makes them more efficient etc. That is, we spend much time in the domain and understand what is good for the user ... While the benefits in the benefits plan are much more narrow. And perhaps a bit artificial.

The engagement and enthusiasm that good stories and personal experience create seems to be pivotal for the motivation to realize benefits. Structured, and perhaps quantified, specifications of benefits do not seem to motivate to the same degree and even seem to demotivate.

Somehow, though, size (quantification) still matters (code: *Small benefits feels meaningless*):

> Sometimes the benefits are small, and that feels meaningless ...

Although the motivation for benefits realization might hinge on engagement and enthusiasm rather than more formal specifications, we will not conclude that the former should be chosen over the latter. However, the findings suggest that it is important to be aware of this possible dilemma, since suggested methodologies both by academia and governing bodies emphasize the explicit, clear and measureable specification of benefits. To make this dilemma explicit, we declare the

following smell, and propose to make a habit of asking the accompanying question:

3) Smell: Dilemma between enthusiasm and formality, Questions to ask:: Are the benefits and the motivation of the relevant stakeholders in harmony? How do the specified and the unspecified benefits relate to each other?

B. Understanding

1) Familiarity: Having experience from previous work that is similar to what lies ahead is reported as a success factor in realizing benefits (code: *Previous experience from similar project aided in realizing benefits*):

> We had a similar project for another internal portal... which had many similarities. The majority of challenges were in areas where the projects were different.

2) Proximity to domain: When benefits span multiple organizations, the ability to recognize possibilities seems to decrease when a stakeholder's distance to the relevant domain increases (code: *Distance between domains reduces understanding*):

> They are further away, so you need more frequent meetings ... they might not have the same semantic understanding of the information provided to them ... It's like using a topographic map to navigate at sea, which can have grave consequences. If you are within the same agency, you usually have a pretty good understanding. If you are within the same sector, you might still have a pretty good understanding. When we start to talk about services that go across sectors, understanding starts to be reduced, and when you move into the private or municipal sector it is even worse.

Both familiarity with similar work and proximity to domain are well-known success factors when it comes to controlling the iron triangle factors cost, scope and time, and caution has been raised toward using data and experience from earlier initiatives that are not very similar [45]. Here, these perspectives appear for benefits realization as well.

3) Smell: Situational differences, Questions to ask:: Do those who need to understand the benefits, the conditions for the benefits, and the relevant situational factors have sufficient understanding? Do we have the necessary conditions (such as time and mindset) and data (on situational factors) to increase our understanding?

4) The ability to understand resistance to benefits realization: When encountering impediments or resistance to benefits realization, understanding the domain where resistance occurs is important in order to evaluate 1) if the resistance is warranted or based on false assumptions and 2) how to mitigate the impediments/resistance.

An example of warranted resistance can be seen in the following excerpt. Here the new process resulted in loss of access to information for a user group. This information was necessary for the users to complete their job assignments. Through dialogue with the users and understanding of the users domain, a solution was found by changing regulations and changing the new software solution so it would provide an aggregated version of the data to the users (code: *Resistance mitigated by providing new functionality*):

> And this brings us to the [name of user group]. We had not thought about them. Losing access to the data from the previous process, they could not complete their responsibility of [responsibility] ... The ministry of health and care services helped us change the relevant regulations, providing us with the legal basis for including the statistics that [name of user group] needed.

Resistance to the new solution can also be unwarranted. One organization that lost access to information when introducing a new software solution saw this as a problem, because the information was important for their tasks. However, it turned out that the process used by the organization was not in accordance with current regulations. Even though representativies from the organization had wanted to continue using the old process when the new software solution was introduced, they learned that this was not an option, without changes to regulations (code: *Resistance mitigated by understanding and sharing information*):

> They were very worried about losing access to [information name] ... We prepared well and invited the [organization name] to a meeting, with our lawyers. In the meeting we explained that the processes they followed today were not in accordance with the relevant regulations. And if they need to continue as before, they need a change in regulations in their sector. So during the meeting they did not get any of what they wanted, but it was still a positive meeting, and they thanked us for informing them and preparing and presented the case well. The key was involvement, and that functioned well.

In both the above examples, understanding – and working to understand – was key to 1) evaluate if the resistance was warranted, and 2) decide how to handle the resistance.

5) Smell: Resistance to realization, Questions to ask:: What resistance to benefits realization are there among stakeholders? How warranted is that resistance?

6) The ability to understand possibilities: The ability to understand possibilities is reported as important for successful benefits realization. A challenge raised in this regards is that many people seem to have a "linear" way of thinking, which limits their perception of new possibilities (code: *Some have a linear way of thinking*).

> We were struggling to get ownership in the line organization. We still struggle with that ... When we have new needs, they don't consider self service to be an option. They recreate their work process in a modern architecture. So if we are talking about processing of [case type anonymized], such as changing the name of a [anonymized object type], they expect a

> more efficient system for registering this ... While in reality, the users can do it by themselves ... It is characterised by not thinking about new ways to do things. A linear way of thinking based on how we used to work.

The ability to recognize opportunities requires personal chracteristics, including competence on digitalization and on the relevant domains (code: *Need digitalization and domain competence to see possibilities*):

> To look up and see the possibilities, then you need a person that is able to look outward, and onward. Understand how data can be used and understand user needs ... You are dependent on a person who can think in terms of digitalization, think new ... And you need to understand the tools ... You need IT competence and you need domain competence in order to understand the user.

7) Smell: Slipping opportunities, Questions to ask:: Do we have the conditions necessary for recognizing new opportunities for benefits realization? Do those who are in position to recognize new opportunities have the necessary understanding of those conditions and the competence of the relevant domains?

C. Project Size

In general, respondents report that it is more challenging to succeed with realizing the benefits from larger projects than from smaller projects. Reasons include that smaller projects require less followup and that it is easier to make people interact when the project is small. Further, larger projects have more tasks that are unrelated to benefits creation, and there are more things that can go wrong in larger projects

Still, smaller projects can suffer from a lack of priority, having key resources who are allocated late or shared with other projects, and difficulties in obtaining assistance.

As the statements related to project size are straight-forward, we do not include excerpts here. However, what seems to be the common denominator mentioned with regards to project size is the challenge to maintaining a focus on benefits.

1) Smell: Loss of focus due to project size, Questions to ask:: Is the project maintaining a focus on benefits realization in the face of organizational size issues, such as overhead and complexity (large initiatives) and lack of priority and visibility (small initiatives)?

D. Dependencies

Dependencies at work outside of the project or within the organization can affect benefits realization negatively.

1) Changes in regulations: Digitalization in the public sector often involves the processing of personal data. In the digitalization projects we studied, it happened that the project uncovered that processes were not defined according to regulations or that the new process required changes in regulations (both of which are exemplified in Section IV-B). Dependencies on changes in regulations puts benefits at risk, especially when there are uncertainties about the regulations.

2) Contributions from other organizations: The observed collaborations have been less focused on contractual agreements and more focused on pragmatic collaboration to realize benefits. An effect of this collaborative basis is that people's and organizations' contributions are based on the different stakeholders' perceptions of benefits, rather than on a set of agreed-upon common benefits to be achieved.

Respondents reported problems with this collaborative basis when other organizations did not contribute with what was necessary. However, one project which seemed especially successful in ensuring contributions from the other organizations described how they worked actively to keep organizations involved (code: *Ensured contribution by keeping organizations involved*):

> This work [ensuring external organizations' contribution] started at day one. When we wrote the mandate for the pilot project, we collaborated with [contributing organization]. And all the contributing organizations were involved in the pilot project. We had defined seven domains and spent a week exploring each of them to understand the situation ... So they had taken part in describing the problem, as much as they had contributed to designing the solution. And then we had them with us. Since then we have had a regular meeting every Wednesday ... to work with the needs, look at and comment designs and user stories, and to test each iteration before final user testing. These meetings has continued even after the project was finished.

The degree of involvement that makes sense for each contributing organization is likely to vary from situation to situation, but being aware of the potential fragility of their involvement and contribution has been raised as a concern.

3) Smell: Lacking commitment, Questions to ask:: Which external parties are we dependent upon and how confident are we in their (continued) involvement and contribution?

E. The Need and Ability to Reach Benefits Recipients

It is often necessary to interact with those who are supposed to benefit from a system (the benefits recipients) to help or make them use the system in their work or life processes. Direct benefits recipients are those who get benefits from the system itself, while indirect recipients get benefits as follow-on effects of the effects that the direct recipients experience. Constellations of those who receive the benefits of a system vary from a few direct recipients, through many direct recipients to a mix of direct and indirect recipients. When there are only a few benefits recipients, spending time on each recipient might not represent a large cost. When the number of recipients is large, the amount of time spent on each recipients is often expected to be low. This can pose a problem when adoption does not go as expected; especially when the diversity among benefits recipients is large. That is, the cost of reaching all recipients is large when there are many, or they cannot be reached directly, such as when benefits recipients are a peripheral part of the process far away from the organization

owning the new process/solution (code: *We cannot get in touch with all recipients*):

> When creating a self service solution, we cannot get in touch with all [anonymized profession], or all citizens of Norway, to make them use the new solution.

1) Smell: Insufficient contact with recipients, Questions to ask:: Do we need to, and do we have the resources to, reach the benefits recipients to ensure benefits realization?

V. Discussion

The benefits of a system impacts work and life processes and are (or should be) rooted in business, organizational and societal goals. Understanding how a system will contribute to those goals – that is, understanding the system's benefits – through shifting and evolving technological, organizational and psychosocial mechanisms is to understand a complex and opaque problem (Section II). According to one school of thought, human beings have evolved to make *good enough* (satisficing [46]) judgements on minimal cues in complex situations [47], [48]. However, due to this complexity and opacity, practitioners often lack data and/or the appropriate verbalizations to back those perceptions and judgements. As a result, actions might be taken late or not at all.

The projects smells that emerged in this study can be seen as empirically-based encapsulations of practitioner insights for ensuring better benefits realization. The smells embody good-enough actions in a complex and opaque environment. Relating again to managerial problem-solving (Section II), the three most important factors for people in identifying a problem – expertise, stance and attention management [28] – can function as impediments to problem detection. While influencing people's expertise and stance can be costly and time consuming, we believe the project smells can be a low cost solution to focusing people's attention on factors that are important for realizing the benefits of software projects. If organizations include project smells for early detection of problems with benefits realization as part of what they keep track of – and pay attention to – the threshold for problem-stimuli to actually be detected may become lower. This should put practitioners in a situation where they can react timely to problem-stimuli.

If adopted in an organization or in a project, project smells could provide important reflection points for software engineers with a legitimacy for concerns raised by those closest to were relevant observations are made. The time horizon for actions based on project smells should not be in the future, at the stratgeic level, but rather, in the short term or immediately.

Smells are not termination indicators. They are indicators that something must be done to make a situation better, but this has to happen on time. Projects who practice benefits management activities *during project execution* seem to be more successful on benefits realization, and also on other success criteria [1]. Brooks famously said: "How does a project get to be a year late? ... One day at a time" [49]. Thinking analogously for benefits realization, the need for day-to-day adjustments becomes pertinent. These day-to-day adjustments can only be done if software engineering teams and their managers understand what is going on. The project smells, we argue, helps teams identify and understand what is going on regarding benefits realization and can help practitioners to identify the right time to take action. Rather than the project smells being binary warning lights that managers should monitor at the cost of everything else, we think people should hold them in mind to guide them when talking with people or otherwise observing their projects.

There are a myriad of recommended sensible actions one can take in development initiatives with the aim to get good results, but it is hard to tell what, of all these things, to do and when to do it, before it is too late. In hindsight, there are often few surprises to what went well or wrong, but the trick is to do something before the fact, and the project smells are an empirically-based contribution to that.

We do not address the follow-up question of what to do for each project smell and who should do it. While this may seem as an omission, trying to list all the sensible ways to react to each smell quickly comes out of hand. Indeed, whereas the smells are generic, the ways to take action must be specific and depend on the details of the initiative's organization and culture. These specific details will also influence how the interaction of events which might lead to a mix of smells.

VI. Limitations

The main threats to validity to the empirical study are construct validity and external validity [50], [51].

A. Construct Validity

Construct validity for the SDI method concerns the extent to which the concepts are well-defined (accuracy) and whether they are validly founded in the data (reliability) [36], [51]. As mentioned, the concept under elaboration in this article ("characteristics of projects that affects the realization of benefits") is one of several concepts elicited in a larger study. Although the reflexive approach does not see a single reviewer at lower levels of coding as threat to validity, scholars versed in the neopositivist tradition might still consider this a threat to construct validity. The SDI method itself has safeguards to heighten validity, even when using one reviewer. Moreover, the fact that many concepts were elicited in the larger study, which demands extensive adjusting of the various concepts to gain a level of integrity and distinction for each concept relative to the other concepts, also gives credibility for accuracy and reliability for the concept under elaboration here. In this study, the concept was also refined further by both authors. In our case, construct validity justifies generalizability, roughly speaking, to situations for which the concept, including the project smells which are derived from the concept, applies.

B. External Validity

This concerns the extent to which the results obtained for the study's sample and situation hold across other samples and situations. The sample is designed, rather than random, in that the

projects were incentivized to perform benefits management. In the outset, this poses threats to generalizability. However, the sample is particularly relevant to the topic of interest, which increases the construct validity of the responses. This is advantageous for conceptual development, which is our aim in this study. Also, the sample is *critical* [52], in that challenges with benefits management, and project smells, that appear in the sample are arguably even more present in non-incentivized settings. On the other hand, external validity may be reduced, since our sample may be biased by special interest in the topic and that the sample is from the Norwegian public sector. To validate the concept and the smells, further studies should be conducted with other samples and in other development contexts.

VII. Conclusion and Further Research

Further studies that observe the use of the smells will tell the extent to which the proposed seven project smells for benefits realization are useful. Observational studies, in similar and related context, will hopefully lead to refinements of the smells and the identification of further smells.

Benefits realization concerns the effects of using a software product, and the idea is that the project smells will facilitate a product focus in projects. Further, it is natural to study the smells in product-centric development, where cross-functional autonomous teams are responsible for the entire lifecycle of functional areas.

The notion of code smells inspired, in form, our notion of project smells. Work on code smells has been ongoing for close to 25 years, and both manual and static approaches to identifying code smells have been applied [53]. Project smells for early detection of problems with benefits realization, as presented here, is not even at the level of the first publication on code smells [19], which have names and labels for a large set of different smells. Although labelling the identified project smells is easy enough, we propose to postpone that exercise, and rather focus on the understanding that lies behind each project smell. That way, the labels (which are less important than the understanding) can come at a later point, when the categories of smells has congealed as a result of further research.

Acknowledgments

The authors are grateful to Magne Jørgensen and Geir K. Hanssen for their contributions in designing and conducting interviews. The authors are further grateful to the interview participants for sharing their time and insights, and to Maurizio Giardini, Eckhart Köppen, Stian Støle, Lene Renneflott, Adam Lawson and Kristoffer Flottorp for their feedback and reflections on the developed concept.

References

[1] M. Jørgensen, "A survey of the characteristics of projects with success in delivering client benefits," *Information and Software Technology*, vol. 78, pp. 83–94, 2016.

[2] S. S. Tanilkan and J. E. Hannay, "Benefit considerations in project decisions," in *Proc. Int'l Conf. Product-Focused Software Process Improvement (PROFES)*, pp. 217–234, Springer, 2022.

[3] D. Baccarini, "The logical framework method for defining project success," *Project management journal*, vol. 30, no. 4, pp. 25–32, 1999.

[4] G. Bradley, *Benefit Realisation Management: A practical guide to achieving benefits through change*. Routledge, 2016.

[5] S. Jenner, *Managing Benefits: Optimizing the Return from Investments*. The Stationery Office, APMG-International, 2014.

[6] C. Lin and G. Pervan, "The practice of IS/IT benefits management in large Australian organizations," *Information & Management*, vol. 41, no. 1, pp. 13–24, 2003.

[7] T. Melton, P. Iles-Smith, and J. Yates, *Project Benefits Management: Linking projects to the Business*. Butterworth-Heinemann, 2008.

[8] M. Payne, *Benefits Management: Releasing project value into the business*. Project Manager Today, 2007.

[9] J. Thorp, *The Information Paradox: Realizing the Business Benefits of Information Technology*. McGraw-Hill, revised ed., 2007.

[10] J. Ward and E. Daniel, *Benefits Management: How to increase the business value of your IT projects*. Wiley, 2nd ed., 2012.

[11] R. Breese, S. Jenner, C. E. M. Serra, and J. Thorp, "Benefits management: Lost or found in translation," *International Journal of Project Management*, vol. 33, no. 7, pp. 1438–1451, 2015.

[12] Infrastructure and Projects Authority (UK), "Guide for effective benefits management in major projects," guidance to practitioners, Infrastructure and Projects Authority, 2017.

[13] J. E. Hannay, H. C. Benestad, and K. Strand, "Benefit points—the best part of the story," *IEEE Software*, vol. 34, no. 3, pp. 73–85, 2017.

[14] C. Larman and B. Vodde, *Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum*. Addison Wesley, 2010.

[15] D. Leffingwell, *Agile Software Requirements: Lean Requirements Practices for Teams, Programs and the Enterprise*. Addison Wesley, 2011.

[16] D. Reinertsen, *Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing, 2009.

[17] J. E. Hannay, H. C. Benestad, and K. Strand, "Earned business value management—see that you deliver value to your customer," *IEEE Software*, vol. 34, no. 4, pp. 58–70, 2017.

[18] M. Haaber and P. Grøhøj, "Benefit points in scrum: A design science study," tech. rep., Dept. of Computer Science, Aalborg University, 2018.

[19] M. Fowler, K. Beck, J. Brant, W. Opdyke, and D. Roberts, *Refactoring: improving the design of existing code. addison*. Wesley Longman Publishing Co., Inc., 1999.

[20] A. P. Snow and M. Keil, "The challenge of accurate software project status reporting: a two-stage model incorporating status errors and reporting bias," *IEEE Transactions on Engineering Management*, vol. 49, no. 4, pp. 491–504, 2002.

[21] S. Petter, "If you can't say something nice: Factors contributing to team member silence in distributed software project teams," in *Proceedings of the 2018 ACM SIGMIS Conference on Computers and People Research*, pp. 43–49, 2018.

[22] J. Andrea, "The case of the missing fingerprint: Solve the mystery of successful end-of-projects retrospectives," *Better Software*, pp. 30–36, February 2007.

[23] G. Meszaros, *xUnit test patterns: Refactoring test code*. Pearson Education, 2007.

[24] B. Van Oort, L. Cruz, B. Loni, and A. Van Deursen, ""Project smells" – Experiences in analysing the software quality of ML projects with mllint," in *Proc. 44th IEEE/ACM Int'l Conf. Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pp. 211–220, 2022.

[25] U. Telemaco, T. Oliveira, P. Alencar, and D. Cowan, "A catalogue of agile smells for agility assessment," *IEEE Access*, vol. 8, pp. 79239–79259, 2020.

[26] M. Cohn, "Toward a catalog of scrum smells," 2003.

[27] E. E. Smith, "Concepts and thought," *The psychology of human thought*, vol. 147, 1988.

[28] G. Klein, R. Pliske, B. Crandall, and D. D. Woods, "Problem detection," *Cognition, Technology & Work*, vol. 7, pp. 14–28, 2005.

[29] D. A. Cowan, "Developing a process model of problem recognition," *Academy of Management Review*, vol. 11, no. 4, pp. 763–776, 1986.

[30] H. Mintzberg, D. Raisinghani, and A. Theoret, "The structure of "unstructured" decision processes," *Administrative Science Quarterly*, vol. 21, no. 2, pp. 246–275, 1976.

[31] R. Chow, K. Christoffersen, and D. D. Woods, "A model of communication in support of distributed anomaly response and replanning," *Proc. Human Factors and Ergonomics Society Annual Meeting*, vol. 44, no. 1, pp. 34–37, 2000.

[32] R. F. Haines, "A breakdown in simultaneous information processing," in *Presbyopia research: From molecular biology to visual adaptation*, pp. 171–175, Springer, 1991.
[33] A. Mack, "Inattentional blindness: Looking without seeing," *Current directions in psychological science*, vol. 12, no. 5, pp. 180–184, 2003.
[34] S. S. Tanilkan, "Benefits management – a study of public sector digitalization projects." https://tinyurl.com/PubSecBM. Accessed: 2024-04-08.
[35] J. Corbin and A. Strauss, *Basics of Qualitative Research*. Sage Publications, 2015.
[36] A. Tjora, *Kvalitative forskningsmetoder i praksis*. Gyldendal Norsk Forlag AS, 2020.
[37] M. Alvesson and K. Sköldberg, *Reflexive Methodology: New Vistas for Qualitative Research*. Sage Publications, 2018.
[38] V. Braun and V. Clarke, "One size fits all? what counts as quality practice in (reflexive) thematic analysis?," *Qualitative research in psychology*, vol. 18, no. 3, pp. 328–352, 2021.
[39] C. O'Connor and H. Joffe, "Intercoder reliability in qualitative research: Debates and practical guidelines," *Int'l J. Qualitative Methods*, vol. 19, 2020.
[40] K. K. Holgeid, M. Jørgensen, D. I. K. Sjøberg, and J. Krogstie, "Benefits management in software development: A systematic review of empirical studies," *IET Software*, vol. 15, 2021.
[41] Direktoratet for Økonomistyring, "Gevinstrealisering – planlegging for å hente ut gevinster av offentlige prosjekter," guidance to practitioners, Direktoratet for Økonomistyring, 2014.
[42] M. Saunders, P. Lewis, and A. Thornhill, *Research methods for business students*. Pearson Education, 8th ed., 2019.
[43] W. Huang, *The Management of Continuous Product Development: Empirical Research in the Online Game Industry*. Springer Nature, 2022.
[44] S. S. Tanilkan and J. E. Hannay, "Projects vs continuous product development – does it affect benefits realization?," in *Proc. Int'l Conf. Advances and Trends in Software Engineering (SOFTENG)*, pp. 20–25, 2023.
[45] S. S. Gautam and V. Singh, "The state-of-the-art in software development effort estimation," *J Software: Evolution and Process*, vol. 30, no. 12, 2018.
[46] H. A. Simon, *The sciences of the artificial*. MIT press, 1996.
[47] G. Gigerenzer and P. M. Todd, *Simple heuristics that make us smart*. Oxford University Press, USA, 1999.
[48] R. M. Hogarth, *Educating intuition*. University of Chicago Press, 2001.
[49] F. P. Brooks Jr, "The mythical man-month (anniversary ed.)," 1995.
[50] T. D. Cook and D. T. Campbell, *Quasi-Experimention: Design & Analysis Issues For Field Settings*. Houghton Wifflin Co., 1979.
[51] W. M. Trochim and J. P. Donnelly, *Research methods knowledge base*, vol. 2. Atomic Dog Pub. Macmillan Publishing Company, New York, 2001.
[52] R. K. Yin, *Case study research: Design and methods, volume 5 of Applied Social Research Methods Series*. Sage, 3rd ed., 2003.
[53] S. Olbrich, D. S. Cruzes, V. Basili, and N. Zazworka, "The evolution and impact of code smells: A case study of two open source systems," in *2009 3rd international symposium on empirical software engineering and measurement*, pp. 390–400, IEEE, 2009.