



# BAM - Backlog Assessment Method

Richard Berntsson Svensson<sup>1</sup>(✉), Tony Gorschek<sup>2</sup>, Per-Olof Bengtsson<sup>2</sup>,  
and Jonas Widerberg<sup>3</sup>

<sup>1</sup> Department of Computer Science and Engineering,  
Chalmers | University of Gothenburg, Gothenburg, Sweden  
`richard.berntsson.svensson@gu.se`

<sup>2</sup> Software Engineering Department, Blekinge Institute of Technology,  
Karlskrona, Sweden

<sup>3</sup> Telenor Sweden, Campus Grasvik 12, Karlskrona, Sweden

**Abstract.** The necessity of software as stand-alone products, and as central parts of non-traditional software products have changed how software products are developed. It started with the introduction of the agile manifesto and has resulted in a change of how software process improvements (SPI) are conducted. Although there are agile SPI methods and several agile practices for evaluating and improving current processes and ways-of-working, no method or practices for evaluating the backlog exists. To address this gap, the Backlog Assessment Method (BAM) was developed and applied in collaboration with Telenor Sweden. BAM enables agile organizations to assess backlogs, and assure that the backlog items are good-enough for their needs and well aligned with the decision process. The results from the validation show that BAM is feasible and relevant in an industrial environment, and it indicates that BAM is useful as a tool to perform analysis of items in a specific backlog.

**Keywords:** Backlog Assessment Method · Agile ·  
Software process improvement · Software process assessment ·  
Case study

## 1 Introduction

The pervasive and necessary presence of software as stand-alone products, but also as central parts of the offering of traditionally non-software products (e.g. ranging from cars to washing machines) has changed the way we have to do software intensive product development [1]. Continuous delivery, flexible engineering methods and ways of working has been one answer to the increased pace and level of software intensive product development. It started off with the agile manifesto in 2001 [2], but speeding up into a myriads of interpretations of “agility” and its realization through the introduction of many agile software development methods (ASD), e.g. Scrum [3].

However, regardless of which development method, software process improvement (SPI) [4] is still important for improving the capabilities of the software

development teams and organizations [5]. Instead of focusing on optimizations as in traditional development, SPI in ASDs has focused on high flexibility and responsiveness that can be standardized both within and across organizations [6]. Hence, the agile perspective changes how to conduct SPI, and requires new SPI mechanisms [5]. Also, improvement and learning is to some extent “built in” to ASDs, e.g., Scrum has the retrospective phase [7].

As a part of our industry collaboration with Telenor Sweden we noticed that one of the main concepts and structures of their agile implementation, namely the Backlog(s), had some challenges. The backlog is among the most, if not the most used agile practice, described as “the heart of Scrum” [8]. Backlogs are the engines for business, planning, and development. However, how do agile teams know if the backlog is good enough and well managed, or if the content of the backlog is appropriate and aligned in relation to the competences working with it and the decision process utilized?

There are guidelines, e.g. INVEST [8] and DEEP [3], of how backlog items should be written and managed. In industry, the backlog items vary greatly, and this is especially true for backlogs that are used for collecting items from multiple sources continuously, and there are seldom any formal checkpoints for ensuring conformity or quality as in traditional software development methods. With great variation of backlog items in the backlog, there is a challenge to align the decision process (people and method for decisions) to be able to handle the varying contents of the backlog. Often items are deferred or delayed as detail is low or abstraction too high, and/or decisions are made based on assumptions as a way to fill the gaps of incomplete information. Therefore, to enable agile organizations to assess backlogs, and assure that the backlog items are good-enough for their needs, and well aligned with the decision process, we developed and evaluated the Backlog Assessment Method (BAM) in close cooperation with Telenor Sweden. BAM was designed to be light-weight, simple to apply, as well as adaptable to the organizations needs and preference - consistent with a method to be used during a retrospective phase for example.

The remainder of this paper is organized as follows. In Sect. 2, background and related work are presented. Section 3 describes the motivation of the development of BAM, while Sect. 4 presents BAM. Section 5 presents lessons learned from using BAM in practice, and Sect. 6 presents the conclusions.

## 2 Background and Related Work

Agile software development methods (ASD) require new SPI methods to fit the context and principles of ASD. One reason for this is that the “traditional” SPI methods have heavy bureaucracy [5], while the agile SPI methods need to be flexible and responsive to local needs, and encourage self-organizing teams [9]. Thus, new challenges for conducting SPI in ASD have emerged [5].

Complying with the agile principles, agile SPI methods have been proposed and evaluated in the literature. In a systematic literature review, Santana et al. [10] investigated which SPI approaches are used for agile SPI. Santana

et al. [10] concluded that there are three main approaches to agile SPI, (1) top-down approaches where the goal is to “fit” agile practices into predefined models, e.g. the agile maturity model [11] and agile-CMMI SPI method [12], (2) agile SPI methods that are based on improving team members’ behavior, which is proposed in the agile manifesto [2], and (3) agile SPI methods that are based on improving practices in order to deliver better software. Moreover, Salo and Abrahamsson [5] proposed an iterative improvement process for conducting SPI within individual agile project teams to improve the development process based on the teams’ experiences and context knowledge, while Ringstad et al. [13] suggest to use diagnosis and action planning to improve teamwork in agile software development. In addition to the agile SPI efforts, there exist many practices in agile methodologies to evaluate and improve current processes. For example, Crystal includes a reflection workshop [14] and Scrum has sprint retrospective [7]. Moreover, Value Stream Map [15] and retrospectives [16] are practices that can be used to evaluate and improve current processes.

Although there are agile SPI and several agile practices for evaluating and improving current processes and ways of working, there are no similar methods or practices for evaluating the most used agile practice, namely the backlog. A well-managed backlog that is well aligned with the decision process can greatly reduce the time for planning meetings and time-to-market. However, if the backlog is ill-managed, used as a dumping ground, and not aligned with the decision process, it may have severe impacts on product development. In particular, since the backlog should provide a centralized and shared understanding of what to build, and in which order to build it [3]. Despite the importance of the backlog in ASD, there are no methods/frameworks for assessing the backlog other than pre-emptive ones. Instead, the literature suggests following guidelines when writing the backlog items (i.e., requirements as user stories) to make sure that the backlog items, once they have been written, follow a certain guideline. One suggested guideline for writing the items is INVEST [8] where the backlog items should be Independent, Negotiable, Valuable, Estimatable, Sized appropriately, and Testable. Once the items have been written and placed in a backlog, it is important to manage, organize, and administrate the backlog items, which is called *grooming* [3,8]. Grooming includes creating and refining details about an item, and to estimate and prioritize them. An example of a guideline that can be followed when Grooming a backlog is DEEP [8].

However, from the process assessment at Telenor Sweden (see Sect. 3.2 for details) we observed that backlogs are frequently used for managing all kinds of information. Often the backlog items vary greatly, and there are seldom any formal checkpoints for ensuring conformity or quality. In addition, obsolete backlog items are common to be found in backlogs in industry [17], meaning that information in form of obsolete requirements is visible in backlogs to a large extent in practice [17]. The obsolete backlog items may lead to, e.g. higher estimates of the items and thus affect the decision-making [18].

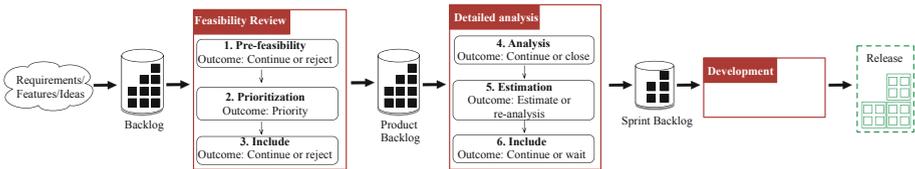
### 3 Research Context: Case Description and Motivation

The development of BAM was prompted by challenges/improvement issues identified at Telenor Sweden. Telenor Sweden’s pragmatic use of their backlog to collect items from multiple sources continuously identified challenges for the organization to handle and process all items in the backlog in an effective and efficient manner, as described in Sect. 3.2.

#### 3.1 Case Company Description

Telenor is one of the world’s major mobile operators with 208 million mobile subscribers. Telenor provides tele, data, and media services in the Nordic, Central, and Eastern Europe and Asia, and have operations in 13 markets, and additionally 14 markets through the ownership of VimpelCom Ltd. Telenor has more than 36,000 employees worldwide, and 1,900 at Telenor Sweden. Telenor Sweden has worked with agile and backlogs for about 10 years.

As a developer of software and services, adopting an agile approach makes sense to avoid rigid front-heavy development. The core scenario is that in the Telenor Sweden case, backlogs are used in multiple stages, which is a common practice in agile software development [8] from an initial backlog where items of large variety are collected, then the items are “sorted” into product backlogs, and finally in a spring backlog, as illustrated in Fig. 1.



**Fig. 1.** A simplified overview of the decision process for items

In each backlog containing items, the items are analyzed and refined, and sometimes broken up/merged/dismissed before progressing to the next backlog, e.g. in feasibility review and detailed analysis phases (as illustrated in Fig. 1) through several decision-points. This way of working joins business analyst/product management level and product planning with subsequent product realization (sprint backlog) levels. This is one of the main goals of becoming a complete agile organization. It is also in line with the general levels of decision-making in agile software development [19]. In agile software development, decisions about products and release plans are made at a *strategic level* (similar to the feasibility review in Fig. 1, while decisions related to project management with the aim to determine the best way to implement the strategic decisions are made at *tactical level* (detailed analysis phase in Fig. 1. Finally, decisions about the implementation are made at the *operational level* (development phase in Fig. 1.

Telenor Sweden’s agile process supports product planning and development and they have good ideas of using their backlogs at different levels which enables other roles, teams, and parts of the organization to work “agile”. However, this creates several challenges, which are detailed below.

### 3.2 Motivation

The case study presented in this paper was carried out at Telenor Sweden in Karlskrona for one of their major product lines (due to confidentiality reasons, no information about the project can be revealed). The overall aim was to identify improvement potential in their development process. Therefore, the first step was to perform a process assessment of their current development process.

*Research Design.* First, brainstorming sessions, meetings to plan the assessment, and selection of which projects and roles to involve in the assessment were conducted. The selection of interview subjects was conducted in close cooperation with a “gate-keeper” at Telenor Sweden. Five interview subjects representing different roles were chosen, 1 process developer, 1 business analysts, 1 product manager, 1 project manager, and 1 operational manager. Data was collected using semi-structured interviews [20] and documents, i.e. archival data collection [21]. The interviews were carried out by one interviewer and one interviewee. For all interviews, we took records in form of written extensive notes. The study of documents was a substantial part of the overall assessment. The analyzed documents included backlogs, formal process descriptions, decision and meeting protocols. All of the extensive notes from the interviews, and the relevant documents were analyzed using content analysis [20].

*Undesired Consequences.* From the process assessment results, it was possible to identify three undesired consequences (UDC) that may stem from incomplete information about backlog items, too abstract backlog items, or having the wrong scope for a particular backlog and point in time. The three UDC are illustrated in Fig. 2, and described in more detail below.

**UDC1: Postponed Backlog Items.** Postponed backlog items refer to items that are not good enough for analysis/decision-making due to incomplete information (see UDC1 in Fig. 2). That is, backlog items with incomplete information are not aligned with the decision process (i.e. decisions at a certain point in time require certain information), and thus the process and the team members involved in a certain decision point cannot do their jobs. This may lead to that backlog items are excluded throughout the entire development process, from feasibility review to development. Postponed backlog items, which are often kept in the backlog, risk becoming obsolete items [17], and can even clog up the backlog. New items are continuously added to the backlog thus inflow is often higher than removal and completion. This may lead to a backlog that is not only cluttered, but also has too much information (items). The mere fact that the postponed items are visible to the decision teams have an impact on the accuracy of the

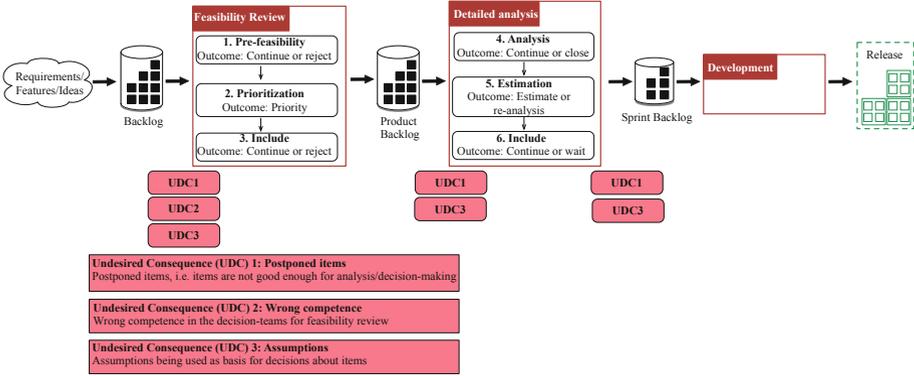


Fig. 2. Three undesired consequences

decisions, estimates, priorities, and analysis [22], where the estimates can be twice as high as the actual effort [18].

**UDC2: Wrong Competence in the Decision-Teams.** If several of the backlog items during the pre-feasibility review (Step 1, Fig. 2) are technically oriented, e.g. related to software architecture, and no architects are present, the feasibility review and its decisions are made without the required competence, or get postponed as more information is requested. This may lead to several problems downstream, not least that assumptions about realization and inaccurate estimates as well as architectural impact can lead to degradation and increased technical debt [23]. If technical expertise is called in to join the decision team the UDC is handled by changing the process and/or team composition. However, the continuous nature of item processing in backlogs would demand that completely dynamic teams with people representing all competences be “on-call” continuously. This may not always be practical, especially not in situations where items range in thousands, many decision meetings process items continuously, and several products are handled by teams distributed over locations.

**UDC3: Assumptions Being Used as Basis for Decisions.** If items are under or over-specified for a particular development phase there is an increased risk of false assumptions (UDC3, Fig. 2) being made, and the rest of the analysis/development phase “downstream” can inherit the issues. The later in the development process this is discovered the costlier to correct, not to mention the analysis effort wasted on the incorrect analysis track. Both in the feasibility review and in the detailed analysis (Steps 3 and 6, Fig. 2), if a backlog item that is under-specified is treated as an item with the right specification level, then the inclusion decision may be wrong. Hence, items with high business value may be excluded from the product/release planning as a consequence of competition of features for resources. During the effort estimation (Step 5, Fig. 2), if the under-specified backlog item is treated as an item with the right specification level, the estimates will be based on inadequate information. Thus, the items are

potentially underestimated, which invalidates projected lead-times and potentially causes missed opportunities such as being first on the market with a new feature, gaining market shares, loss of sales, and lower revenue.

## 4 Backlog Assessment Method

This section gives an overview of BAM and the guidelines of how it can be used to plan, execute, and analyze the results of the backlog assessment.

BAM offers a visual analysis and grading of backlog items. The visualization serves as a focal point for the stakeholders to collectively discuss if the backlog items have appropriate detail, size, maturity, and most importantly, if the decision-making process within the development organization can handle the variation of the items in the backlog. Thus it is important to realize that BAM does not suggest a level, rather allows the span to be estimated based on the real items in the backlog, then compared to the capacity of the organization and how the items are intended to be used. BAM has the benefit of using concrete artifacts (actual backlog items) as a base for the assessment, rating items on a scale based on four main perspectives that were identified in collaboration with practitioners at Telenor. The four main perspectives, which are described in detail in Sect. 4.4, are: Scope, Abstraction/Level, Maturity, and Detail.

The decision of which backlog to assess, selection of a representative sample of items from the backlog, team members to involve - as well as the details of the actual assessment are detailed in Fig. 3. The following sub-sections describes each step of BAM in more detail.

### 4.1 Step 1: Identify Relevant Backlog

An organization may have several products, thus several product backlogs, or different type of backlogs for the same product, as seen in Step 1, Fig. 3. Having multiple backlogs at several different levels is not an uncommon way to organize the work and requirements in industry. In particular for large organizations where the work requires decisions on multiple levels to select what to realize in the next phase/sprint. The case company Telenor has, in essence, three types of backlogs for a product, one backlog before the feasibility review, one product backlog, and one sprint backlog as illustrated in Fig. 1. In the first backlog all incoming requirements/ideas/features (called items from hereon in) are initially gathered before a feasibility review takes place. An item is then, either dismissed or it progresses to the product backlog, and then to the sprint backlogs. Each of these backlogs contains items that are inherently specified in different ways. That is, the scope in the initial backlog is wider with less details, and in the later backlogs the information is refined and detailed. This is not unlike any engineering process where requirements (items) are refined and evolve as they progress through the organization, from potential idea to on-queue for development. Weather or not there are one or several types of backlogs, Step 1 in BAM is to select a backlog to assess. A general tip here is, not to see the selection as finite as you could study several backlogs over time.

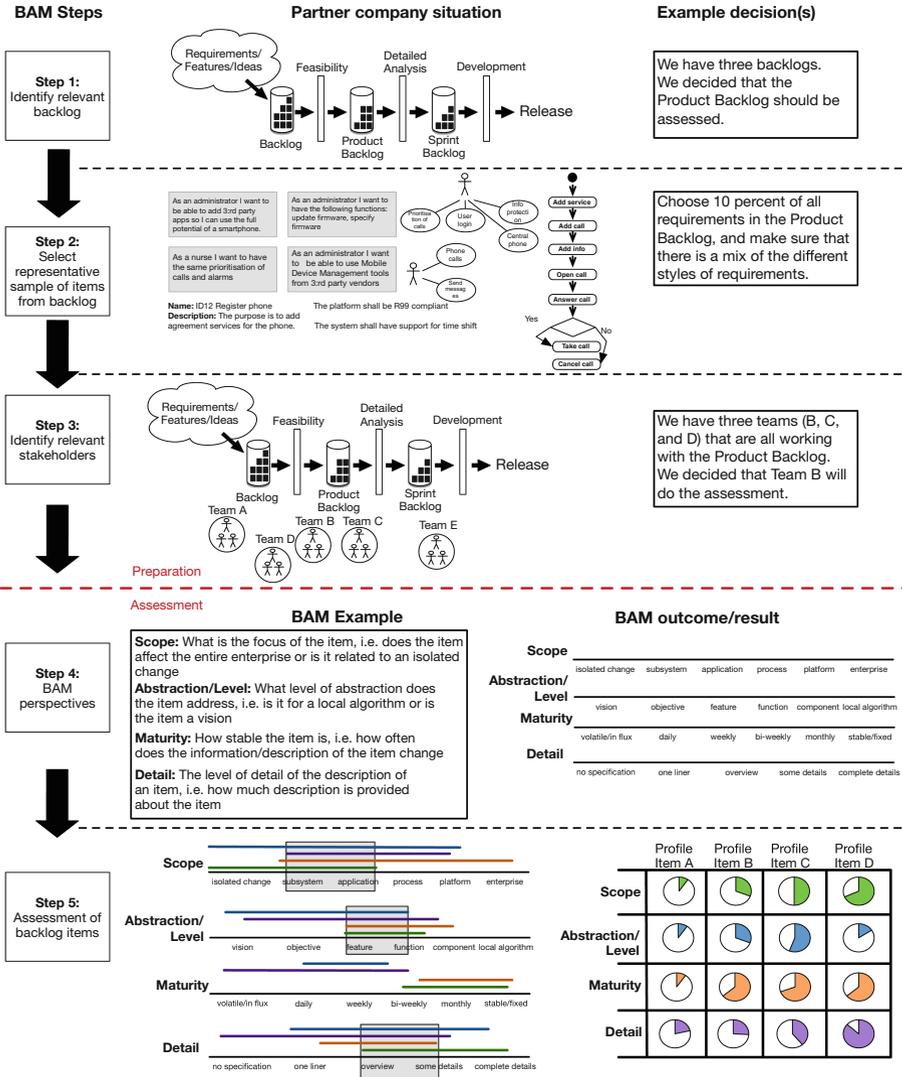


Fig. 3. The five main steps in the backlog assessment method (BAM) (Color figure online)

### 4.2 Step 2: Select Representative Sample of Items from the Backlog

A backlog consists of items that should be included in the product or action taken in relation to it [3]. Regardless of what items that are present in a backlog, they should be written as user stories [8]. However, in practice, not all items that should be included in the product are included in the backlog, and not all items are written as user stories. Instead, it is common to write items using natural lan-

guage [24,25]. Telenor Sweden uses several different styles of writing/specifying requirements/backlog items, including use-case diagrams, use-case scenarios, and natural language, as seen in Step 2, Fig. 3. This is not surprising as different specification styles are appropriate at different levels in the decision chain as the abstraction levels differ, and the items are used for different purposes [26].

To have a representative assessment of a backlog, it is important to include items written in all of the existing specification styles that are present in the backlog. Also, it is important to have a good and manageable sample size of items to be able to perform a simple and easy assessment of a backlog - it is better to perform several small assessments than one big one. Whether or not there are items written in one or several specification styles, Step 2 in BAM is to select a representative sample of items from the backlog. To select a representative sample, 10% of all the items, which in the case company was about 100 items, was considered appropriate. However, if the backlog contains thousands of items, then 10% may be too many items. In this case, it is better to select between 50 and 100 items as this size is manageable from an effort perspective (taking one day). In this case representativeness of the selection comes second to the scalability of the assessment.

### 4.3 Step 3: Identify Relevant Stakeholders

The next step is to identify the relevant stakeholders, which may include the team(s) working with the backlog, and/or any other role(s) that use the items in the backlog for, e.g. decision-making, planning future products/sprints, estimations, coordination with other teams/products, or prioritization. The identification of stakeholders could be a simple one-to-one mapping, one backlog and one team working with the backlog, or there could be multiple teams working with the same backlog, (Step 3, Fig. 3). At Telenor there is usually one team with various roles from different departments/areas working with the backlog in the feasibility phase, while other teams work with the product and sprint backlogs. That is, it is not the same team/roles that work with the items from potential ideas to on-queue for development. This is an important characteristic of the case in hand, but also for several organizations that try to use backlogs as a coordination mechanism outside development projects. Multiple roles, multiple teams, representing one or several parts of the organization and/or several products might be relevant to involve in the assessment. Regardless of how many teams/roles that are working with the selected backlog, Step 3 in BAM is to identify the relevant stakeholders for the assessment of the backlog. During the assessment of a backlog at Telenor Sweden, four stakeholders were selected to perform the assessment. It is a good start to start with 4–6 stakeholders to get a first quick overview of the backlog. It should be observed that in our case some stakeholders had multiple roles and thus were involved in several decision points.

#### 4.4 Step 4: BAM Scales

At the core, BAM consists of four *BAM perspectives*, as seen in Step 4, Fig. 3. These are used to grade individual items in the backlog. The four *BAM perspectives* are described in below.

*Scope.* Does the item focus on an isolated change, or is the scope the entire enterprise? The scope of items in the initial backlog is generally wider, e.g. having the scope of the entire enterprise; while in the later backlogs (e.g. product or sprint) the scope often narrows - a natural consequence of refinement and break down. From a scope perspective, it is important that the stakeholders working with a backlog item can conceptualize the item in relation to the length of the sprint. Otherwise, they may have difficulties to break down the item [27]. The scope is different as breakdown of all items directly would result in wasted resources as many items put in the initial backlog that are compared and then passed on refined, or also often dismissed as other items are relatively more important to realize.

*Abstraction/Level.* Does the item address a local algorithm, a feature, or a vision? The *Abstraction/Level* of an item can differ even if the *Scope* is the same. For example, if an item has the *Scope* of a sub-system, the item can address a local algorithm for the sub-system, or it can address the future vision of the same sub-system.

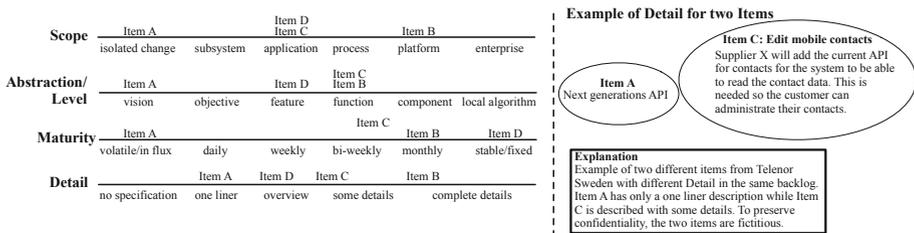
*Maturity.* How often does an item change? One of the “tools” used in agile to embrace change is actually the backlog itself, which should accommodate easy change to items. Albeit change is to be embraced - maturity is connected to change. In an initial backlog (pre-project, where items are collected for e.g. a feasibility review) item may change on a daily basis. However, stability should increase in later backlogs - even if change is still a reality and has to be embraced.

*Detail.* What is the level of specification of an item. Items in a backlog should have different levels of detail depending on what backlog they are specified in. However, the closer to realization (sprint backlog) an item comes, generally the more detail is present to reflect the analysis and choices made for its realization. Greater detail in later backlogs allow team members to know what they are committing to [27]. However, even in the early backlog it is important to have adequate details about the items (good-enough for decisions and prioritization) otherwise it may result in wrongly dismissed or selected items. Too much detail is also undesirable as it would constitute waste. BAM does not suggest a “correct” level, rather it offers a visual evaluation of current state of items in a backlog, and gives practitioners the ability to judge if the level is appropriate, or not, given their needs in relation to a specific decision point and backlog.

#### 4.5 Step 5: Assessment of the Backlog

In the fifth and final step of BAM, the identified stakeholders perform the assessment. Each stakeholder individually estimates items by grading them using each

of the four perspectives from Step 4. The assessment is performed using predefined templates as seen in Step 4, Fig. 3. The template can be used in two ways, (1) using one template for each item, or (2) adding several items to the same template. The case company used one template for several items (see Fig. 4, Items A - D) by placing the item's ID on the scale (to preserve confidentiality, the two examples of detail for Items A and C are anonymized in terms of feature detail). For example, Item A has the *Scope* of an isolated change, the *Abstraction/Level* is graded as vision, the *Maturity* of the information for Item A is in-flux, and the *Detail* of Item A is specified as a one-liner.



**Fig. 4.** Example of an assessment of two backlog items

The reason for adding the item's ID in the template is to be able to perform a sanity check of the assessment. At Telenor, the sanity check was used for two main reasons. First, to have concrete examples of items when presenting the overall assessment makes it easier to understand and discuss the assessment. Second, the items were used as input if one or more stakeholders had completely different assessments of the same item. Thus, the provided item was used to calibrate the overall assessment. A notable observation is that the actual estimation itself served as a learning experience and coordination effort between different roles and competences, in essence building a shared understanding of not only the perspectives themselves, but more importantly of the needs of fellow co-workers. Thus details in an item that was considered as unnecessary by one stakeholder was important for another. This allowed for an understanding to be built, as well as the realization that one person's "waste" could be another person's critical information.

When all stakeholders have completed their assessment, an overview of the results is constructed (Step 5, Fig. 3). The constructed overview comes in two parts. First, the overview shows the total estimated span of items in the backlog for each of the four perspectives combined with the overlap of all stakeholders. At Telenor Sweden, the four stakeholders had different opinions about the span of items in their backlog (see Step 5, Fig. 3). Each of the colored lines represents one stakeholder. For example, one stakeholder believed that the *Scope* of the items span from an isolated change to the platform (blue line), while another stakeholder believed that the *Scope* spans from an isolated change to the application (green line). The grey boxes represent the shared view among the stakeholders

for each scale. Second, the assessment results can also be used to create “typical profiles” of items that exist in the same backlog, as seen in Step 5, Fig. 3. Four typical item profiles were identified in one backlog at Telenor. For example, one typical profile (profile Item A, Step 5 in Fig. 3) was items with a narrow *Scope*, wide *Abstraction/Level*, with constantly changing information (*Maturity*) that was specified with little *Detail*. In the same backlog, other items (profile Item D) had a wider *Scope*, more *Mature* information that was specified in greater *Detail*.

## 5 BAM Case Study Evaluation

As BAM was completed, it was set to be evaluated in an industrial environment in a real development project. The idea was to use BAM for an actual backlog and backlog items in order to evaluate its steps and components (e.g. the four BAM perspectives), and to check if BAM could help in assessing backlogs as a process assessment activity in order to help agile teams to continuously reflect on how to become more effective, efficient and coordinated.

*Research Design.* The selection of backlog and practitioners for participating in the evaluation was conducted in cooperation with the case company. Seven practitioners representing different roles were selected. The roles chosen were: 1 Process developer, 2 Business analysts, 2 Product managers, 1 Project manager, and 1 Operational manager. The selected practitioners identified 100 backlog items that were used as a representative sample of items from the backlog (Step 2 in BAM, Fig. 3). Then the practitioners used the template for BAM perspectives (Step 4 in BAM, Fig. 3) to perform the assessment of the backlog items (Step 5 in BAM, Fig. 3). After the practitioners had used BAM, data - in terms of BAM’s usability and usefulness - was collected. Semi-structured interviews [20] were used to ask questions about how BAM was perceived by the practitioners. The interviews were carried out by one interviewer and one interviewee. For all interviews, we took records in form of written extensive notes. The extensive notes were analyzed using content analysis [20].

### 5.1 Lessons Learned

Through the evaluation of BAM with Telenor Sweden, we observed experiences summarized below as lessons learned.

*Ease of Use.* In general, the practitioners at Telenor found that BAM is easy to understand, and that the five steps of BAM make sense for assessing a backlog. The practitioners explained that the steps of BAM are helpful and easy to follow, and in particular the provided examples for each step (Fig. 3). In addition, BAM does not seem to take too much time to use in practice. An assessment of about 100 items for one backlog with 4–5 practitioners did not take more than a day, and if the number of items is decreased to 30–40 items, then the time spent on the assessment is only 3–4 h, which could be done on a regular basis by individual teams without formalizing it and getting “permission” or a budget.

*Identification of Gaps.* Using BAM as a tool to perform analysis of items in a specific backlog, and the decision process associated with decision points tied to the backlog, enabled us to identify gaps. This in turn gave input to either put some criteria as to good-enough item analysis and specification, and/or changes to the decision process. Often slight modifications of both are needed as per observations at Telenor Sweden. This gave potential to remove the likelihood of undesired consequences (as explained in Figure 2), but also speed up processing and decrease the clutter in the backlog of lingering items. BAM does not introduce the “correct” levels, rather, allows an estimation of span of the items in a backlog according to the four perspectives - which in turn can be compared to the decision making process and the capabilities of a team responsible for a specific backlog. Just as extensive technical detail might not be necessary for a quick feasibility analysis, abstract business goals are too abstract for a technical development team. BAM thus supports:

1. How items should be specified (good-enough) for a specific backlog,
2. what level of information and investment into analysis is needed for a team,
3. how the decision-making process should look like to be aligned with the backlog items to enable reduced time for planning and analysis until an item can be dismissed or selected for evolution into a subsequent backlog.

BAM will not in-itself solve undesired consequences (e.g. UDC1 - UDC3). However, by assessing the backlog items it is possible to get an overview of the level of information in the backlog, and get the ability to see the typical item profiles and level of span of said items. This can then be used to identify potential issues with the alignment between the backlog and the decision-making process. If the decision-making process is not aligned with the backlog, there is a risk that the teams either spend more time and resources in analyzing the backlog items to make sure that all of them have a similar level of information, or that the team may want to change the decision-making process and criteria to handle all kinds of item profiles by, e.g. changing team member profiles or adding more team members. BAM can also successfully be complemented with a Value Stream Mapping [15] activity to gauge the delays and times of items in play, and the changes as backlog item details and decision processes are tweaked.

## 5.2 Limitations

Selection bias (*construct validity*) [21], in terms of selecting participants to interview in relation to the process assessment and the evaluation of BAM, is a threat to this study. Selection bias is always a threat to all studies where the participants are not fully randomly sampled. The threat is that only participants with a negative attitude of the current processes and ways-of-working, and/or having a positive attitude towards BAM were selected. To minimize this threat, the participants were selected based on their experiences and roles by a “gate-keeper” at Telenor Sweden. That is, the researchers did not influence the selection of the participants for the interviews, nor did the researchers influence the selection of

which participants/teams that applied BAM in practice. Incorrect data (*internal validity*) [21] is a validity threat to all empirical studies. To minimize this threat, the researchers had the opportunity to validate, both during the interviews and after the interviews by contacting the participants about the answers and interpretations of the answers, which minimizes the risk of misunderstandings (i.e. collecting wrong (incorrect) data). *External validity* [21] is related to the ability to generalize the results from this study beyond the case company. Although Telenor Sweden is a large company developing software and services, it is not representative for all large software developing organizations. However, the aim of qualitative studies is not to generalize beyond the studied setting, instead, qualitative studies aim to explain and understand the phenomenon that is studied [20]. However, some of the issues introduced as motivation behind BAM (see Sect. 3.2) may be generalized for organizations that are using or introducing ASD methods. Moreover, from the concepts, practical application, and evaluation of BAM (as described in Sects. 4 and 5) may give an overview of the specific situation of Telenor Sweden, thus allowing other organizations to judge similarity and potential for BAM being relevant for them.

## 6 Conclusions

The Backlog Assessment Method was developed and evaluated in close cooperation with Telenor Sweden. BAM is based on the identified needs in industry and the observations that existing agile practices and agile SPI methods only provides guidelines of how backlog items should be written, and guidelines of how to keep the backlog groomed. However, there are no methods or practices for assessing the current backlog and the backlog items. BAM addresses this issue, aiming to enrich the overall picture of the information in the backlog.

As part of BAM's development, evolution, and refinement, the complete version of BAM was evaluated at Telenor Sweden with seven industry practitioners using a real backlog and actual backlog items from a product. The evaluation was performed to assure BAM's applicability in industry, and that the model is useful as part of agile SPI improvements for individual teams. During the evaluation at Telenor Sweden, the results show that BAM is feasible and relevant in an industrial environment, and the evaluation results indicate that BAM is useful as a tool to perform analysis as to items in a specific backlog, and the decision process associated with the backlog. The main benefit of using BAM is the use of concrete artifacts as the base for the assessment to give visual indications as to how the backlogs are used. The value is in getting results from an evaluation as discussion and decision support material to the practitioners working with the backlogs, thus BAM does not prescribe any correct level or processes.

The next phase that will be undertaken in the validation and evolution of BAM, as described in this paper, is further evaluations in industry in different domains where the long-term effects, in terms of benefits and challenges of using BAM, need to be investigated to fully validate its feasibility and scalability.

**Acknowledgements.** The work and results presented in this paper were performed and produced in close cooperation between industry and academia. The authors would like to thank everyone involved in the development and evaluation of BAM at Telenor Sweden.

## References

1. Petersen, K., Wohlin, C.: A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *J. Syst. Softw.* **82**, 1479–1490 (2009)
2. Agile Alliance: Manifesto for agile software development (2001). <https://www.agilealliance.org/agile101/the-agile-manifesto/>. Accessed 5 Jan 2019
3. Cohn, M.: *Succeeding with Agile: Software Development Using Scrum*. Addison-Wesley, Boston (2009)
4. Aaen, I., Arent, J., Mathiassen, L., Ngwenyama, O.: A conceptual MAP of software process improvement. *Scand. J. Inf. Syst.* **13**, 81–101 (2001)
5. Salo, O., Abrahamsson, P.: An iterative improvement process for agile software development. *Softw. Process. Improv. Pract.* **12**(1), 81–100 (2007)
6. Nerur, S., Balijepally, V.: Theoretical reflections on agile development methodologies. *Commun. ACM* **50**, 79–83 (2007)
7. Schwaber, K.: *Agile Project Management with Scrum*. Microsoft Press, Washington DC (2004)
8. Rubin, K.S.: *A Practical Guide to the Most Popular Agile Process*. Addison-Wesley, Boston (2013)
9. Allison, I.: Towards an agile approach to software process improvement: addressing the changing needs of software products. *Commun. IIMA* **5**(1), 67–76 (2005)
10. Santana, C., Queiroz, F., Vasconcelos, A., Gusmao, C.: Software process improvement in agile software development: a systematic literature review. In: 41st Euromicro Conference on Software Engineering and Advanced Applications, pp. 325–332 (2015)
11. Patel, C., Rumachandran, M.: Agile maturity model (AMM): a software process improvement framework for agile software development practices. *Int. J. Softw. Eng.* **2**(1), 3–28 (2009)
12. McCaffery, F., Pikkariainen, M., Richardson, I.: AHAA - agile, hybrid assessment method for automotive safety critical SMEs. In: 30th International Conference on Software Engineering, pp. 551–560 (2008)
13. Ringstad, M.A., Dingsøy, T., Brede Moe, N.: Agile process improvement: diagnosis and planning to improve teamwork. In: O’Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) *EuroSPI 2011. CCIS*, vol. 172, pp. 167–178. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-22206-1\\_15](https://doi.org/10.1007/978-3-642-22206-1_15)
14. Cockburn, A.: *Crystal Clear: A Human-powered Methodology for Small Teams*. Wesley, Stoughton (2005)
15. Khurum, M., Petersen, K., Gorschek, T.: Extending value stream mapping through waste definition beyond customer perspective. *J. Softw.: Eval. Process.* **26**(12), 1074–1105 (2014)
16. Derby, E., Larsen, D.: *Agile Retrospectives: Making Good Teams Great!*. Pragmatic Bookshelf (2006)
17. Wnuk, K., Gorschek, T., Zahda, S.: Obsolete software requirements. *Inf. Softw. Technol.* **55**(6), 921–940 (2013)

18. Gren, L., Berntsson Svensson, R., Unterkalmsteiner, M.: Is it possible to disregard obsolete requirements? - An initial experiment on a potentially new bias in software effort estimation. In: 10th International Workshop on Cooperative and Human Aspects of Software Engineering, pp. 56–61 (2017)
19. Aurum, A., Wohlin, C., Porter, A.: Aligning software project decisions: a case study. *Int. J. Softw. Eng. Knowl. Eng.* **16**(6), 795–818 (2006)
20. Robson, C.: *Real World Research*. Blackwell (2002)
21. Runeson, P., Höst, M., Rainer, A., Regnell, B.: *Case Study Research in Software Engineering: Guidelines and Examples*. Wiley, Hoboken (2012)
22. Eppler, M.J., Menigs, J.: The concept of information overload: a review of literature from organization science, accounting, marketing, MIS, and related disciplines. *Inf. Soc.* **20**(5), 325–344 (2004)
23. Tom, E., Aurum, A., Vidgen, R.: An exploration of technical debt. *J. Syst. Softw.* **86**(6), 1498–1516 (2013)
24. Savolainen, J., Kuusela, J., Vilavaara, A.: Transition to agile development - rediscovery of important requirements engineering practices. In: 18th IEEE Requirements Engineering Conference, pp. 289–294 (2010)
25. Berntsson Svensson, R., Olsson, T., Regnell, B.: An investigation of how quality requirements are specified in industrial practice. *Inf. Softw. Technol.* **55**(7), 1224–1236 (2013)
26. Lauesen, S.: *Software Requirements: Styles and Techniques*. Addison-Wesley, Boston (2002)
27. Berczuk, S.: Back to basics: the role of agile principles in success with a distributed scrum team. In: *Agile Conference*, pp. 13–17 (2007)

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

