P. Jönsson and C. Wohlin, "Using Checklists to Support the Change Control Process -- A Case Study", Proceedings of Software Engineering Research and Practice in Sweden, Umeå, Sweden, 2006.

Using Checklists to Support the Change Control Process—A Case Study

Per Jönsson*
per.jonsson@bth.se

Claes Wohlin
claes.wohlin@bth.se

School of Engineering
Blekinge Institute of Technology
PO-Box 520, SE-372 25, Ronneby, Sweden

ABSTRACT

Change control is crucial in a software development organisation, as changes arrive during all stages of development. Changes that are introduced in the software at the peril of individual developers may affect aspects of the software such as stability, usability, and overall quality. Typically, change proposals are handled in a change control process, where the implications and impact of the change are analysed. In order to support this type of analysis, we have created a checklist-based process support instrument, the purpose of which is to allow for objective and systematic analyses. We present an approach for adapting generic checklists along three axes: process, domain, and roles, as well as characteristics of the resulting support instrument. We also describe two evaluations of the change control process, of which the latter is designed with the stepwise character of the process in mind. We discuss some preliminary evaluation results, and in the light of these conclude that the use of checklist-based process support seems promising so far.

1. INTRODUCTION

During the development of a software system, the initial scope and conditions are bound to change for several reasons. Such changes, which, for example, can be caused by modifications in hardware or software platforms, new customer demands, and project overruns, typically require corresponding changes in the software. In order to avoid a situation where the system falls apart because of changes that should have been dismissed (or implemented differently), it is important to have a change control process in place [10, 11]. A change control process allows changes to be controlled properly, which leads to a more stable product.

Typically, a change control process involves a number of steps through which a change proposal (or change request) is formally evaluated and, based on factors such as change consequences, system impact, and implementation cost, either accepted for implementation or rejected. The evaluation of a proposed change is known as *change impact analysis*, an activity that is about finding the implications of a change before it is implemented. Definitions of the impact analysis concept vary; their commonalities are about consequences of a change, whereas differences lie mostly in levels and characteristics of the artefacts of interest [1, 12, 17].

In this paper, we discuss *change request control*, which we see as the totality of all evaluative aspects of the change control process. In other words, it encompasses screening of and discussions about a proposed change, but also the change impact analysis activities. It should also be noted that impact analysis that takes place as part of change request control differs in characteristics from impact analysis made before implementing the change. More specifically, the objective of change request control is to be able to make a decision about the inclusion or exclusion of the proposed change, and the granularity of impact analysis only needs to be fine enough to allow this. Before implementing a change, on the other hand, the impact analysis needs to be more detailed in order to take into account system impact and possible ripple effects of the change [1].

In this paper, we present the following:

- The design of an approach used for adapting a set of generic checklists, intended for analysing a proposed change, to suit the specific use context.
- The characteristics of a checklist-based process support instrument, which is the outcome of the aforementioned adaptation in the context of the change control process at Ericsson AB, Sweden.
- The design and execution of two process evaluations, of which one has been performed and one is ongoing, targeted at the change control process.

The adapted process support instrument (henceforth termed *PSI*) comprises a number of checklists, each containing items that are relevant for the sake of controlling and analysing a proposed change. Our basic expectation is that the use of the PSI leads to increased process efficiency, as it supports

*Corresponding author.

SERPS'06 October 18-19, 2006, Umeå, Sweden

its users in making informed decisions. The main difference between the two evaluations described is that one is specifically tailored for the stepwise character of the change control process.

According to Hales and Pronovost, who report on the successful use of checklists in fields such as aviation, product manufacturing, and critical care, the main purpose of using checklists is to reduce errors or to increase adherence to standards [5]. Their view is that checklists are an important cognitive aid—more formal than Post-It notes but less formal than a defined protocol—that can provide guidance to the user and thereby improve the outcome, or performance, of the activity. While the use of checklists is common in the field of software inspections, studies have reported discouraging results as to their usefulness when looking for defects (see, for example, [13] and [16]). However, we see inspection with the goal of defect detection as different from using a checklist to reduce errors in performing an activity, whereby the latter is the scope of this paper. In other words, we do believe that checklists can provide good process support.

The paper is organised as follows. Section 2 presents study background, including research context and related work. The design of the approach for checklist adaptation is described in Section 3, together with the evaluation designs. Thereafter, the outcome of the adaptation in our case (i.e., characteristics of the PSI) as well as preliminary results from the evaluations are presented in Section 4. Section 5 contains a brief discussion of the implications of the results, together with some notes on industrial adoption and threats to validity. Finally, Section 6 summarises the paper.

2. BACKGROUND

This section describes the background of the study. We present the research context for the study, as well as relevant related work. We also discuss the change control process, previous work, and aspects of using checklists for process support.

2.1 Research Context and Objective

The studied company is a specific development unit (DU) of Ericsson AB, Sweden. Ericsson develops software systems for the telecommunications industry, an industry that has been rapidly evolving over the past years, and continues to do so. Projects run at the DU are typically long, around 12–18 months, and thus very sensitive to changes. In order to deal with the many changes, it is imperative that the change control process functions efficiently and as intended.

The overall goal of the research presented here has been to provide a lightweight means of supporting the change control process. The research questions are:

RQ1: Is checklist-based process support an effective way of improving the change control process?

RQ2: How can we measure the efficiency of the change control process in an effective way?

2.2 Related Work

Ramzan and Ikram discuss decision-making in requirements change management, and stresses the importance of making informed decisions [14]. They argue that the existence of relevant information, such as stakeholders affected by a change, cost versus benefit, and associated risks, is crucial when making firmly based decisions. Our standpoint is that a checklist can help the user taking into account different types of relevant information.

Wiegers provides a set of checklists for supporting the analysis of change impact [18]. His checklists are meant to be further customised when used (which is what we have done and describe in this paper). Like Wiegers, Hass promotes the use of checklists for doing impact analysis (the term she uses is *consequence analysis*), and presents a customisation of Wiegers' checklists [6].

Keil et al. study the use of checklists for risk identification [9]. From their perspective, the purpose of a checklist is to allow the user to identify more risks and make better decisions regarding project continuation. One of their findings is that the use of checklists do lead to more risks found, and they argue that checklists may increase the user's sensitivity to risks (and thereby likelihood of finding risks).

Brykczynski performed a survey of 117 checklists[1] for software inspection, including not only ones suitable for for finding defects, but also ones meant for more general review [2]. Among other things, he argues that checklists for non-code work products tend to be general in nature and should focus on *consistency*, *correctness* and *completeness*.

Despite the fact that checklists in software inspection are common, some studies report that using checklists is not the best means for finding defects. Thelin et al., for example, have compared usage-based and checklist-based reading, and found usage-based reading to be superior over checklist-based reading with respect to finding faults [16]. Porter and Votta compared three methods for finding defects in software requirements specifications, and found that the checklist method was outperformed by the other two, ad hoc and scenario [13]. As we see the use of checklists for finding defects as different from what we have done in this study, we do not consider the otherwise discouraging results from the software inspection field to be an inhibitor.

As pointed out earlier, definitions of change impact analysis differ, ranging from "the assessment of a change, to the source code of a module, on the other modules of the system" by Turner and Munro [17] to "the evaluation of the many risks associated with the change, including estimates of effects on resources, effort and schedule" by Pfleeger [12]. This means that change impact analysis can be seen on different levels.

Change impact analysis is often a manual activity, carried out through interviews with knowledgeable developers [7]. However, it is also possible to use more systematic methods such as consulting design documentation, using slicing techniques, and performing dependency and/or traceability analysis [7]. As mentioned in the introduction, the work behind this paper is concerned with the entire process of controlling a change request, rather than only the change im-

[1]None of these were intended for change requests, though.

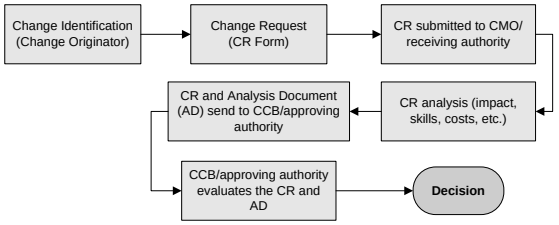

Figure 1: Steps in the change control process according to Leon [11]

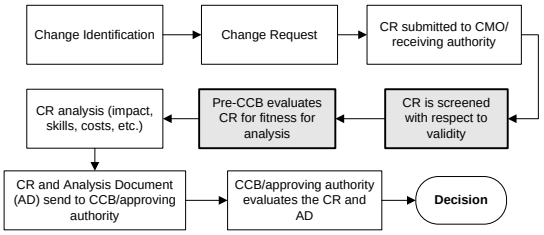

Figure 2: Steps in the change control process at the studied company. Additional steps in comparison to Leon's process are highlighted.

pact analysis activity. Thus, change impact analysis methods or techniques, such as the ones mentioned, should be used in conjunction with checklists in order to obtain the information necessary to "check off" items in the checklists.

The distinction between the two stages of change impact analysis discussed in the introduction is also made by El Emam et al. [4]. They distinguish between *preliminary analysis*, the purpose of which is to allow the appropriate board to approve (or reject) the change request, and *detailed change analysis*, the purpose of which is to identify all necessary modifications resulting from the change.

2.3 Change Control Process

Leon, like many others, highlights the problem of uncontrolled change, and argues that change management and control solves problems of communication, sharing of data, multiple maintenance, and simultaneous updating [11]. He provides a generic process to use for change management and control, here illustrated in Figure 1. The Configuration Management Officer (CMO) receives the change request from the author, and reviews it for clarity and completeness. If necessary, the change request is sent back to the author for clarification. Once the change request is complete, it is passed on in the process for analysis and finally decision [11].

Leon points out that the process he presents is generic and typically varies from company to company [11]. At the studied company, the change control process is similar, albeit with some extra steps. Figure 2 shows the steps in this process.

As can be seen, the process features a specific step for screening change requests (also referred to as filter meeting), the purpose of which is to ensure that only valid change requests get passed on in the change control process. This reduces the stress on the process. Furthermore, the process has a pre-CCB meeting, which acts as another tollgate in the process. At the pre-CCB meeting, it is decided whether or not the change request is fit for detailed analysis prior to the final decision in the CCB.

As with Leon's process, the initial part of this process is iterative, meaning that the change request, if deemed invalid in the screening step, can go back to the author for clarification and re-work.

A number of process roles are defined at the studied company. The *CR issuer*, who can be anyone, writes the change request initially. The change request is submitted to the *CR receiver* (corresponding to Leon's CMO role [11]), who works tightly in the screening stage to ensure the validity of the change request. The *CR analyser* (of which there typically are several) is responsible for analysing the change request. The *implementation responsible* is responsible for taking care of the change request after it has been accepted.

2.4 Previous Work

In a previous study (see [8]), we examined issues of change request handling and change impact analysis at the studied company. In the data collection phase, during which a number of interviews were performed, two observations were made:

1. The top three forces for the analysis of change impact were *gut feeling*, *experience* and *competence* (as opposed to models and other documentation).
2. Change requests had become a very effective means of communication within the development projects (due to their importance; nobody would dismiss one). Unfortunately, this had led to a certain amount of abuse, as change requests were used for other things than proposing changes.

The first observation may not come as a surprise, but the actual consequences are nonetheless important to consider. Any analysis made based solely on subjective forces is bound to be person dependent and, in the worst case, inconsistent with what someone else would produce. This points towards a need for support during the change control process. For example, during change impact analysis, support could result in more objective and well-founded analyses, and also decrease the risk that something is missed during an analysis.

The latter observation means that an early screening is necessary in order to discard all invalid change requests. Partly as a result of the observation, such a screening step has been introduced at the studied company (see Section 2.3), yielding greater control over the items entering the change control process. The screening step, being a point of evaluation in the process, also calls for support that could ease the process of deciding whether or not a particular item is indeed a proper change request that should be handled as such.

During the interviews, process support, preferably in the form of checklists, was actually requested. Checklists had been used before, but were mainly project specific and never promoted to a more general level. A combination of the need for and the explicit request for checklist-based process support were among the main motivators for the work described in this paper.

The top organisational issues found in the interview study were that (i) system impact is often underestimated or overlooked when analysing; (ii) that analyses generally are incomplete or delayed; and (iii) that parties affected by a change request are often overlooked in the analysis [8]. We believe that the issues are a consequence of subjective impact analysis, and thus argue that checklist-based process support should be geared towards these aspects.

2.5 Checklists

This section describes general aspects of using checklists, and provides an overview of the generic checklists that have been the basis for the PSI discussed in this paper.

2.5.1 Checklist Aspects

It is relevant to discuss advantages and disadvantages of using a checklist as process support. A checklist is an effective way of making sure that nothing is omitted or forgotten, but its use can of course lack in regularity and thoroughness. It is common to have a checklist when grocery shopping, at least if the number of items to buy is more than a few. Without such a checklist, it is is easy to forget some items. This is a consequence of how the human brain works; we can only keep track of a limited amount of items at the same time (see [3]). Similarly, a pilot has to go through a checklist before takeoff (and in various other situations), in order to ensure that safety procedures are properly gone through [5].

The implications of not using the checklist are in these two examples vastly different. Forgetting something when grocery shopping may ruin a meal, whereas a pilot's negligence could be a matter of life and death. The consequences of forgetting something when analysing the impact of a software change lies somewhere in between, and can mean everything from hardly anything to project overruns, unhappy customers and lost revenue.

Apart from the obvious cognitive support described above, a checklist has the following advantages when used for the sake of process support:

- It is a non-intrusive instrument for assisting in activities that are otherwise driven by subjective forces such as experience. Since such subjective forces should not be underestimated, a good support instrument should complement rather than replace them.

- A checklist can be used to various extents, depending on the situation. For example, a large complex change request would entail a thorough walkthrough of the checklist, whereas a simpler change request would entail a more brief walkthrough. Thus, the use of a checklist can be very flexible.

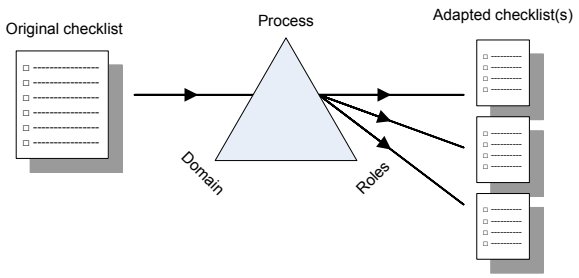

Figure 3: Illustration of checklist adaptation.

- Adding an objective part to an activity, a checklist allows for consistency between performers of the activity.

A possible disadvantage of a checklist, on the other hand, is that when people get used to the checklist, they may stop using it because they believe they know it by heart. Either they do, in which case the checklist has become a successful part of the organisational culture, or they do not, in which case the situation is back to as it was before introducing the checklist.

2.5.2 Wiegers' Checklists

When creating the PSI, we decided to base it on existing material, namely the checklists proposed by Wiegers [18]. Wiegers presents two checklists for change impact analysis, plus one form for effort calculation. The first checklist, hereafter referred to as the *implications checklist*, consists of items that correspond to possible implications of a proposed change, and Wiegers argues that the analyst can use it for the sake of understanding the change better [18]. An example item is: "What are the possible adverse side effects or other risks of making the proposed change?" The second checklist, hereafter referred to as the *impact checklist*, aids in identifying software elements affected by the proposed change. An example item from this checklist is: "Identify any changes required in build files or procedures."

Wiegers is, to our knowledge, one of the few authors of requirements engineering books that discusses change impact analysis (as part of the change control process) more in detail and actually provides some sort of support for performing it. A set of checklists provided by Hass are adapted from Wiegers' checklists, but have coarser granularity [6]. However, her use of Wiegers' checklists suggest that they are indeed meaningful in this context, and it is appropriate to use them as starting point.

3. RESEARCH METHODOLOGY

In this section, we explain how Wiegers' generic checklists were adapted to the specific needs of the studied company, and describe the designs of the two process evaluations.

3.1 Checklist Adaptation

Wiegers suggests to modify his checklists to better suit the projects in which they are used [18]. This is not uncommon; Brykczynski mentions that none of the checklists in his survey should be used "as is", but rather be tailored to specific needs [2].

Our adaptation approach is illustrated in Figure 3. Basically, adaptation was performed along the following three axes:

Process The process axis represents the current change control process at the company. Process adaptation means in our case to split the original checklists into several smaller checklists according to the steps in the process. This way, the items in each step are specifically tailored to the needs of that step.

Domain The domain axis represents the application domain and, more generally speaking, the software development context at the company. Adapting to the domain should be done to avoid the omission of domain-specific aspects that need to be considered when dealing with a proposed change.

Roles The roles axis represents the roles of the persons involved in the change control process. Role adaptation means to distribute the items in a checklist into different areas of responsibility, and assigning the responsibilities to roles. This way, different aspects of analysing a proposed change are dealt with by the persons who are most suitable.

The three axes where selected based on needs identified at the company; we do not wish to assert that these three are the only ones, or the best ones. Other thinkable axes are software architecture and software requirements. However, these are on a different level of abstraction than the three we have used, and could tie the resulting PSI too close to a specific system.

The adaptation was performed through meetings with the various stakeholders of the change control process, such as project managers, configuration managers, technical coordinators, and test leaders. The objective was to include persons relevant from the perspectives of Wiegers' original checklists (i.e., change implications, detailed impact analysis, and effort calculation).

The outcome of the adaptation was the actual PSI, consisting of a set of checklists tailored for the domain and mapped to the relevant change control process steps. The role adaptation, however, was not made on all the resulting checklists, but only on the one for detailed impact analysis (i.e., identification of affected software elements). The reason was that this checklist was more diversified than the remaining checklists. The adaptation outcome is further discussed in Section 4.1.

3.2 Process Evaluation

In order to be able to measure the effect of introducing the PSI at the company, it was decided to perform an evaluation for measuring the current process as a baseline. The evaluation form should preferably be reusable by the company at any later stage to measure the process health either routinely or in response to some process change. Thus, the evaluation should be both relevant with respect to the PSI, but also with respect to the process in general.

Two evaluations are described here. The first one was indeed performed, but a number of disadvantages where highlighted afterwards. Based on these, a second evaluation was designed in order to obtain a better view of the state of the process.

3.2.1 Evaluation 1

As mentioned, an objective of the evaluation was to allow it to be reused later on. Thus, it was decided to create a lightweight form that could be filled in quickly but still cover the most relevant aspects of the process. As a consequence of this, evaluation 1 was entirely quantitative.

Each item on the form was a statement about the change control process, to which the respondent could express level of agreement on a scale from strongly disagree to strongly agree. This is known as the *summated rating* (or Likert) approach [15]. According to Robson, items developed using this approach can be interesting to the respondents. Thus, the chance of a high response rate can be assumed to be high. Robson advises that the statements should be both positive and negative, and that extreme statements should be avoided [15]. The following considerations were made when creating statements for the evaluation form:

- Statements should express the expected process state rather than a perfect process state. In other words, a statement about quality should not aim for highest possible quality, but rather reasonable or acceptable quality. The motivation for this is that aiming for a perfect process in a large development organisation where processes are adapted and tweaked anyway is rather futile.
- Despite Robson's advice on using both positive and negative statements, all were made positive (in the sense that strong agreement was the desirable answer). The reason for not having negative statements was to avoid the risk of a "not" or any other negation being missed. Since the company staff are busy in general, such a mistake could happen easily, and the evaluation validity would be jeopardised.

With respect to what to cover in the evaluation, we decided to include aspects that (a) are important from the point-of-view of an efficient change control process; (b) we knew from earlier studies could be problematic; and (c) can be affected by the use of the PSI. The included aspects are given below, and the full list of statements can be seen in Table 1.

Change request quality An efficient change control process is dependent on high-quality input. In the previous study about change request and impact analysis issues, unclear change requests were seen as an important issue [8].

Decision easiness Since the change control process is very decision intense, decision-related aspects are relevant to include. The easiness of a decision depends on the availability of input, and should be a good indicator of process efficiency.

Aspect	Statement
Change request quality	In general, the overall quality of a change request is satisfactory
	In general, the time it takes for a change request to be clear is acceptable.
	In general, the number of times a change request must be completed with additional information is acceptable.
Filtering/screening	The relevant aspects of a change request are in general taken into consideration during a filter meeting.
Decision easiness	In general, the extent to which the impact analysis fields of a change request are filled in is acceptable.
	In general, the information available from the impact analysis is complete enough to make a proper decision.
Decision certainty	Change request decisions are in general made with a satisfactory level of certainty.
	Change request decisions are correct in most cases.
Analysis time	The delays between the steps in the CR analysis process are of acceptable length.
	The subprojects/nodes generally perform their impact analyses in time for making decisions.

Table 1: Aspects and statements in evaluation 1. Note that CR analysis process in fact refers to the change control process.

Decision certainty Decision certainty is dependent on the quality of input. As with decision easiness, decision certainty is seen as an indicator of process efficiency.

Filtering/screening The screening of change requests is a separate decision activity in the process. As it is a fairly recent addition, it was decided to distinguish it from the other decision activities.

Analysis time Based on discussions with company staff, it was clear that the process was seen by many as being very time-consuming. Note that analysis here refers to the entire change control process, not only the impact analysis activities.

All of the aspects above represent situations that arguably should improve with the use of the PSI.

Some disadvantages of the first evaluation were highlighted after it was performed. First, using one form for the entire process was suboptimal; different parts of the process could not be distinguished. Second, and related, some confusion arose as to which part of the process the decision aspects easiness and certainty referred to.

3.2.2 *Evaluation 2*

In response to the disadvantages of the first evaluation, a second, more detailed evaluation was designed. In particular, we tried to take into account the stepwise character of the change control process, and direct the evaluation to the corresponding process roles and decision fora.

For the sake of this evaluation, we outlined the change control process as a chain of steps, each with a specific process role. We argued that each step in the chain can receive input from the previous step, provide output to the next step, or both. Also, we recognised that certain steps are explicit decision points. Based on this model, we formulated three labels for the process roles:

A **producer/contributor** creates, refines or adds to the change request. Appending new information, such as an analysis, is a contribution. Making a decision or passing on the change request also counts as a contribution.

A **receiver** gets the change request as input for further work such as verifying appropriateness, making decisions and adding information.

A **decision maker** is part of a decision meeting where it decided what to do with the change request.

Each process role was assigned one or more of these labels, and the set of labels determined the contents of the evaluation form. In other words, we customised the evaluation form depending on the process role characteristics.

The contents of the evaluation form was decided based on the input/output model described above. It was argued that it would be difficult for someone to assess the quality of his or her own output in very much detail, whereas it would be simpler to assess the quality of input (i.e., someone else's output). Thus, for evaluation of output quality, the aspects *confidence* and *effort* were chosen, and for evaluation of input quality, the aspects *completeness*, *self-containedness*, *detail level*, and *overall quality* were chosen.

For each label, a number of statements were given just as in the first evaluation. Each statement was formulated in a positive way, so that strong agreement would be beneficial.

In order to get more information about process efficiency and improvement details, a number of qualitative questions were added to the evaluation. Another reason was to allow the respondents to be more expressive in relation to their answers on the quantitative questions/statements.

The quantitative part of the evaluation form is shown in Table 2, and the qualitative questions are shown in Table 3. In terms of expected outcome, the quantitative part of the evaluation can not only be used to determine process health, but due to its adherence to the input/output character of the process also to find discrepancies in the process. More specifically, a mismatch between the quality of output from

Target	Statement (Generally speaking...)
Producer/contributor	...I have high confidence in the quality of my output.
	...the effort I spend on generating my output is reasonable.
Receiver	...the actual completeness of a CR (i.e. the extent to which all fields are filled in) is satisfactory.
	...the actual self-containedness of a CR (i.e. the extent to which the CR can be understood without consulting external sources) is satisfactory.
	...the actual detail level of a CR is satisfactory.
	...the overall quality of a CR is satisfactory.
Decision maker	...it is easy enough to make a decision for a CR.
	...a CR decision can be made with sufficient certainty.
	...the responsiveness of parties providing input for decision making is satisfactory.
Everyone	...the actual turnaround time (from create to accept) of an accepted CR is reasonable.

Table 2: Quantitative part of evaluation 2

- In your role as *X*, which are your main tasks?
- In your role as *X*, please describe the organisational support you get.
- Please describe the most crucial problems, if any, you have encountered in your role as *X*.
- Could the CR handling process, with respect to your role as *X* or otherwise, be made more efficient? How?
- Do you have anything else you want to add in relation to the aspects covered above?

Table 3: Qualitative part of evaluation 2. *X* in each item corresponds to the process role of the person receiving the evaluation.

one step and the quality of input in the next step may indicate a weak spot in the process (or its use). Such a weak spot may, for example, owe to lack of support in the producing step (resulting in subjectivity), or unrealistic expectations in the receiving step, or a combination thereof.

3.2.3 Subject Selection

The selection of subjects for the checklist adaptation and for the first evaluation was mainly governed by recommendations from key personnel at the studied company. For the checklist adaptation, we tried to select representatives from project management, the development organisation, configuration management, requirements management, and technical experts. In the first evaluation, we wanted to evaluate within one of the currently ongoing development projects, and consequently asked the project manager to identify relevant persons involved in the change control process.

In the second evaluation, subjects were rather selected to represent the process roles. The selection was based on recommendations, both from management and from already identified subjects. This is a sampling strategy similar to what Robson denotes as *snowball sampling* [15]. Many of the process roles are held by a handful employees. Put differently, most of these employees have several roles in the change control process. This means that a sample of any of the process roles is close to the population, but also that it is difficult to find more than one subject per process role, given that each person should participate in the evaluation from one perspective only.

4. RESULTS

This section presents the outcome of the checklist adaptation, as well as some results from the evaluations performed.

4.1 Checklist Adaptation Outcome

As pointed out before, the original checklists by Wiegers were adapted along three axes: *domain*, *process*, and *roles*. For the original checklists, see [18]. The outcome of the adaptation in our case is presented below in terms of characteristics of the resulting PSI.

4.1.1 Process Adaptation

The process adaptation resulted in a set of checklists, each one tailored for a specific step or phase in the process (see Section 2.3). With respect to individual items, this adaptation resulted mainly in deletion or modification of items. The outcome of the process adaptation is summarised below, including the overall characteristics of the checklists.

A *CR preparation* checklist was created to support the initial change request creation. Thus, the checklist should be used by the CR issuer. This checklist contains the fewest number of items of all checklists, mainly because anyone is allowed to create a change request. but not anyone has experience in doing so. Thus, a large checklist at this point could potentially "scare off" change request issuers. In terms of Wiegers' implications checklist, items about baseline conflict and reject consequence are included here. Added items cover self-containedness as well as completeness with respect to the change request form. The latter serves the purpose of increasing the chance that a change request is properly filled in already from the beginning.

A *CR screening* checklist was created to support the change request screening activity. As such, it contains items for self-containedness, correctness, and consistency. However, it also retains items about baseline conflict and reject consequence from the implications checklist. Furthermore, it contains an early impact analysis, but on the level of configuration items. It can be used by the CR receiver before it is passed on to the screening step, and also at the actual screening meeting. Size wise, this checklist is equivalent to the CR preparation list. The main reason for keeping this checklist small is that items of interest at this early stage are primarily related to change request structure rather than detailed analysis.

A *CR prestudy* checklist was created to support the pre-CCB step of the process. The pre-CCB determines whether or not a change request can be passed on to detailed impact analysis. This checklist is more extensive than the previous two, and retains from the implications checklist items for baseline conflict, reject consequences, side effects, quality attributes affected, and project plan impact. It adds items for relationship with pending changes, addition of third-party products, and relationships and impact on configuration items (more detailed than the CR screening checklist).

A *CR analysis* checklist was created to support a general, high-level analysis of change implications. This checklist is intended for use in the analysis phase before the CCB meeting, as a high-level complement to more detailed impact analysis. It retains many of the implications checklist items, such as items for technical consequences, testing, prototyping, project plan, product cost, and lost effort if the proposed change is rejected. A few items have been added, taking into account support and supply activities (since it is not uncommon to overlook these) as well as the need for unfamiliar hardware.

Finally, a *CR technical analysis checklist* was created based on Wiegers' impact checklist. The items in this checklist are not on the form "What is...?", "How much...?", or "Does the...?", but rather "Identify..." and "Estimate...". The purpose of the checklist is to allow for a more detailed impact analysis than the general CR analysis checklist. In a project with several development teams/sub projects, this checklist should be used separately by each team. Compared to Wiegers' impact checklist, the adapted checklist merges items together, such as impact on user interface and documentation into one item, and impact on different types of software entities into one item. The rationale behind this is, as was argued in Section 1, that the impact analysis at this stage (before the change request actually has been accepted) does not need to be very detailed. Added items concern domain-specific items (see below), required design competence, configuration management, and test environments/types. There is also a "best before date" item, signaling to the receiving authority how long the estimated impact (especially in terms of time and resources) is valid. Should the change request decision take a very long time, the estimated impact may become outdated.

It should be mentioned that the process adaptation did not only take into account the different process steps and their characteristics, but also related process documentation. For example, the process and the main tool used both prescribe which information and attributes that must be provided for a change request.

4.1.2 Domain Adaptation

The domain adaptation resulted mainly in the addition of checklist items, and affected only the CR analysis and CR technical analysis checklists. Generally speaking, the domain adaptation concerned the following aspects:

- *Hardware*, as the studied company develops systems rather than only software. In other words, the customers buy prepackaged hardware/software solutions, for which the company has a whole responsibility. Consequently, it is imperative that the lifecycle and characteristics of the hardware used are fully known.
- *Licensing*, again as the company develops entire systems. Thus, licensing of external products must be taken into account.
- *Third-party software*, for the same reason.
- *Whole-world distribution*, as the customers of the studied company are not within one single country, and therefore differences in rules and regulations are important to consider.

4.1.3 Role Adaptation

Role adaptation was performed only for the CR technical analysis checklist. The reason was to distribute the analysis effort among different roles, both to reduce stress and to account for the fact that the items in themselves already are targeted at different roles. Normally, a single analyst would turn to persons with different roles for support in the analysis, but here the responsibilities are explicitly assigned from the beginning.

The role-based division of checklist items also entails a prescribed order in which the checklist items should be dealt with. First, the technical coordinator identifies system impact (both software and hardware), licensing and third-party aspects, and required design competence. The configuration manager then identifies impact on configuration items. Next, the test coordinator estimates the impact on test cases and test strategies. Finally, the project manager summarises and estimates impact on various project-related plans.

4.1.4 Effort calculation

Wiegers' provides an effort calculation form that can be used to calculate the required effort for realising the proposed change. In discussions with the studied company, it was decided that a detailed effort calculation may not be worthwhile. Prior to the change request decision, what is relevant in terms of effort is magnitude (e.g., 100 hours vs. 1000 hours) rather than an exact figure. Thus, the effort calculation form has been disregarded to any further extent so far.

4.2 Evaluation 1

The first evaluation was sent to six persons, having the roles of project manager, test coordinator, technical coordinator, CCB responsible, requirements manager, and screening responsible within a development project running at the time. Five of the six responded.

The summated rating (Likert) scale used in the evaluation allows us to calculate both an overall process health and the health of individual aspects in the evaluation. The health of the process with respect to a set of items can be calculated using the following formula, where p is process health, x is the number of items, y is the number of respondents, c_k is the count over all items and respondents for item score k (i.e., number of 1s, 2s, etc.), and l and u represent the lower and upper scale limits, respectively (in our case 1 and 5): $p = ((\sum_{k=l}^{u} c_k) - x * y * l)/(x * y * (u - l))$.

The formula normalises the summated item ratings in the range from the lowest possible rating sum to the highest possible rating sum, thereby yielding a result between 0 and 1. Transforming the result into a percentage gives us an idea of the health of the process, where 100% means that no improvements (in the evaluated aspects) are necessary.

In our case, the evaluation results indicated that the following two aspects (in the order presented) need to be addressed in a process improvement effort:

- *Analysis time* was seen as troublesome, meaning both that the delays between process steps need to be shortened, and that the impact analysis activity needs to be performed faster.
- *Change request quality* was seen as too low, indicating that the initial stage of the process needs to be addressed specifically. Improved change request quality would of course also shorten the process lead time, as less time would have to be spent on clarifying and completing the change requests.

4.3 Evaluation 2

The second evaluation is ongoing, but some results have been received already. The evaluation was adapted and sent to persons representing the following roles and steps of the change control process at the studied company: CR issuer (3 persons), CR receiver (1), pre-CCB member (1), filter meeting member (1), CCB member (2), and implementation responsible (1). The CR analyser role has not yet been addressed in the evaluation. The results shown here are based on responses from six persons, covering the roles of CR issuer, CR receiver, filter meeting member, and implementation responsible.

As pointed out in Section 3.2.2, the results from the second evaluation can be used both to determine process health, and to find discrepancies in the process. Furthermore, as the evaluation is more detailed than the first one, a process health value of interest can be traced to a specific part of the process. For example, where evaluation 1 could show the general quality of change requests, evaluation 2 allows us to distinguish change request quality in early stages of the process from that in late stages of the process.

Calculating process health is done as in evaluation 1. If an overall health value is desired, the process roles can simply be ignored, such that item scores are counted across roles. For example, in order to determine the overall process health with respect to quality of change requests, the items for change request completeness, self-containedness, detail level, and overall quality (see Table 2), for all receiving roles, should be used. However, to specifically learn about the health with respect to initial change request quality, only the receiving role CR receiver (and possibly filter meeting member) should be used.

If the process health for a receiving role is known, it is interesting to look at the process health for the corresponding producing role. For example, based on confidence in output for the CR issuer role, we could calculate a health value that can be compared to that of the change request quality-related items as seen by the CR receiver role. A mismatch between two health values obtained like this indicates that there is a discrepancy in the process that should be dealt with.

Some preliminary results, both quantitative and qualitative, from the second evaluation as performed at the studied company are presented below:

- In general, people are confident in what they produce, and spend reasonable effort on generating their output. However, as seen both from CR receiver and filter meeting member perspectives, the change request quality could clearly be improved. Thus, there is a mismatch regarding change request quality in this early stage in the process.
- The process health with respect to decisions in the early screening step is in par with the result from evaluation 1.
- The change request turnaround time could be improved, but the answers are on the positive side of the scale. In particular, people seem more optimistic about this time aspect than the ones measured in evaluation 1 (i.e., delays in the process and responsiveness of analysing parties).
- In terms of organisational support received, all respondents highlighted support by other project members and the project organisation in itself. Support by any kind of process instrument was not mentioned.
- Communication with strategic product management can be difficult. Change requests that are not seen as strategic are sometimes dismissed, only to be issued again later on when there is a stronger strategic urge. This causes unnecessary work in the organisation.

In terms of improving process efficiency, no pattern can be recognised so far. However, one respondent pointed out that an iterative change request approach, where a change request is gradually completed with more information, allows for good process flow, since not too much effort has been invested upfront should a change request be rejected early on.

5. DISCUSSION

In this section, we briefly reflect on the evaluation results presented in Section 4.

The purpose of evaluation 1 was to be able to assess a process health. Process health is seen as a simple and easily understood indicator of the state of the process. In recurring evaluations, process health values can be plotted, and trends in the process state can be detected. Looking at specific aspects of the performed evaluation, the results indicated that both analysis time and change request quality are points of improvement, whereas the other aspects are more healthy. A problem with this first evaluation was that it was not tailored to the change control process, and some confusion arose as to which steps in the process were targeted.

The second evaluation was created to take into account the different steps in the process. As such, it is suitable for pinpointing discrepancies in the process, where the output of one step does not match the expected input to the following step. The preliminary results actually show such a case, where the initial change request quality could be better according to receivers, while the producers are confident in their output. Interestingly enough, one CR issuer also argued that an iterative approach to writing change requests results in an efficient process. However, this is somewhat conflicting with the apparent high expectations in the receiving end; an iterative approach would not result in more complete and self-contained change requests.

The second evaluation also contained some qualitative questions. While the preliminary results do not allow for a full analysis, the issue of difficulty in communicating with strategic product management was raised; sometimes non-strategic change requests are dismissed although they should not be. A possible remedy for this is to focus harder in the initial phase of the process on the long-term strategic consequences of both accepting and rejecting the change request.

We believe that checklist-based process support is an important part of process improvement with respect to the improvement issues identified in the evaluations. The results of the second evaluation shows that no process support of this kind or similar exists at the company today.

5.1 Industrial Adoption

As described in Section 2.4, the request for checklist-based process support was raised already in the previous study on change request handling and impact analysis [8]. While working on the adaptation of the checklists, great interest has been shown by process stakeholders. Early on, an internal screening of the initial checklists was initiated inside the company, and several of the stakeholders have commented on the checklists for the sake of improving them.

The PSI has also been partially adopted in the change control process, currently for the pre-CCB step. Furthermore, the company has expressed interest in using something similar also for requirements management, as this process is similar to the change control process. The outcome of the adoption, in terms of further required adaptation and general opinions, has not been investigated yet. However, the intention is to repeat the second evaluation later on to see if anything has improved.

5.2 Validity Threats

Study validity is normally divided into *internal*, *external*, *construct* and *conclusion* validity [19]. Threats to the validity of the study, and counteractions taken, are outlined below.

A study has construct validity if it measures what was intended to measure [19]. A threat to this type of validity is that the evaluations presented (in particular the second, as it is the one of primary interest) may not cover the right aspects, and may not be suitable for measuring process health in general and checklist effects in particular. To void this threat, we strove to base the evaluation aspects on findings in previous studies. In the second evaluation, we also discussed the contents and input/output characteristics of the evaluation form with research colleagues.

A second threat to construct validity is that the term *quality* used in the evaluations has not been defined. However, we argue that this is not necessary. Even if people have different notions of what quality means, an indication of low quality in a process needs to be followed up. Quality has to do with what people expect, so different perceptions of quality reflect different expectations, and an objective of every process is to satisfy all of its stakeholders' expectations. Furthermore, when an evaluation is recurring, the relative quality is of more importance than the absolute quality at any time.

A study has internal validity if the measured effects can be attributed to changes made by the researcher [19]. A threat to internal validity is that the second evaluation is ongoing in parallel to adoption at the company. We do, however, believe that the adoption is not yet so far gone that it would affect the evaluation in a negative way.

A study has external validity if it is possible to generalise the results outside of the study [19]. While this is a single case study, and the selection of subjects has not been entirely random, we consider the checklist adaptation to be successful so far. Thus, the methodology in the study should be generalisable. Also, the input/output approach used in the second evaluation seems promising, and should also be generalisable.

6. SUMMARY

Our contribution in this paper is threefold: (1) an approach for adapting a set of generic checklists for analysing a proposed change, to the context of their use; (2) the characteristics of a process support instrument—the outcome of the adaptation in our case; and (3) a process evaluation approach that takes into account the input/output characteristics of the steps, as well as points of decision making, in the change control process.

The generic checklists were adapted along three axes—process, domain, and roles—in order to create a process support instrument that could cater for all relevant aspects and stakeholders of the process at a studied software development company. The checklists in the resulting instrument cover all phases of change request control, including detailed estimation of system impact.

We have designed and executed two process evaluations. The first evaluation was used to determine the health of the change control process, and resulted in the identification of two important improvement issues: analysis time and change request quality. The second evaluation was designed to overcome some disadvantages of the first evaluation, primarily that it was not tailored for the steps in the process. In the second evaluation, the process steps were seen as having input, output, and/or decision character. By comparing input and output quality, we should be able to pinpoint discrepancies in the process in addition to determining process health. The second evaluation also covered some qualitative aspects of the process, such as problem identification and determination of improvements. While the results are not yet complete, we have already been able to identify preliminary

points of improvement.

The checklist-based process support has been partially adopted at the studied company, and the reception has been positive. We consider this to be favourable with respect to both the checklist adaptation approach and the resulting process support instrument. We cannot yet answer research question RQ1 (Is checklist-based process support an effective way of improving the change control process?), as we will need to perform a follow-up evaluation to see the effects. When it comes to research question RQ2 (How can we measure the efficiency of the change control process in an effective way?), we believe the answer lies in the second evaluation approach. However, we need to finish the evaluation before we can have a final answer.

6.1 Future Work

The most imminent future work is the completion of the second evaluation, and the execution of a follow-up evaluation in order to measure the effect of the PSI. This must be done so we can answer the research questions. Open research questions that have been raised in this work are:

- Can the described checklist-based PSI be used successfully also in the requirements management process?
- How well does checklist-based process support perform in comparison to other approaches, such as tool automation?
- Are there more relevant axes (than process, roles and domain) to be used in checklist adaptation? Is there something other than need that determines which axes that are relevant to look at?

7. ACKNOWLEDGMENTS

This work was partly funded by The Knowledge Foundation in Sweden under a research grant for the project "Blekinge—Engineering Software Qualities (BESQ)" (`http://www.bth.se/besq`). We would like to express our gratitude to Martin Byggeth for help with adapting the generic checklists for the process axis, and to Patrik Berander and Piotr Tomaszewski for helpful advise when designing the second evaluation. Finally, we thank the anonymous reviewers for valuable comments that have helped in improving the paper.

8. REFERENCES

[1] S. A. Bohner and R. S. Arnold. *Software Change Impact Analysis*. IEEE Computer Society Press, 1996.

[2] B. Brykczynski. A survey of software inspection checklists. *SIGSOFT Software Engineering Notes*, 24(1):82–89, 1999.

[3] J.-L. Doumont. Magical numbers: The seven-plus-or-minus-two myth. *IEEE Transactions on Professional Communication*, 45(2):123–127, 2002.

[4] K. E. Emam, D. Höltje, and N. H. Madhavji. Causal analysis of the requirements change process for a large system. In *Proceedings of the International Conference on Software Maintenance*, pages 214–221, Washington, DC, USA, 1997. IEEE Computer Society.

[5] B. Hales and P. Pronovost. The checklist—a tool for error management and performance improvement. *Journal of critical care*, 21(3):231–235, 2006.

[6] A. M. J. Hass. *Configuration Management Principles and Practice*. Addison-Wesley Professional, 2002.

[7] P. Jönsson and M. Lindvall. Impact analysis. In C. Wohlin and A. Aurum, editors, *Engineering and Managing Software Requirements*, chapter 6. Springer-Verlag, 2005.

[8] P. Jönsson and C. Wohlin. Understanding impact analysis: An empirical study to capture knowledge on different organisational levels. In *Proceedings of the International Conference on Software Engineering and Knowledge Engineering*, pages 707–712, Taipei, Taiwan, July 2005.

[9] M. Keil, L. Li, L. Mathiassen, and G. Zheng. The influence of checklists and roles on software practitioner risk perception and decision-making. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, page 229b, Washington, DC, USA, 2006. IEEE Computer Society.

[10] D. Leffingwell and D. Widrig. *Managing Software Requirements—A Unified Approach*. Addison Wesley, 1999.

[11] A. Leon. *A Guide to Software Configuration Management*. Artech House Publishers, 2000.

[12] S. L. Pfleeger. *Software Engineering—Theory and Practice*. Prentice-Hall, intl. edition, 1998.

[13] A. A. Porter and L. G. Votta. An experiment to assess different defect detection methods for software requirements inspections. In *Proceedings of the 16th international conference on Software engineering*, pages 103–112, Los Alamitos, CA, USA, 1994. IEEE Computer Society Press.

[14] S. Ramzan and N. Ikram. Making decision in requirement change management. In *First International Conference on Information and Communication Technologies*, pages 309–312, Karachi, Pakistan, 2005.

[15] C. Robson. *Real World Research*. Blackwell Publishing, 2002.

[16] T. Thelin, P. Runeson, and C. Wohlin. An experimental comparison of usage-based and checklist-based reading. *IEEE Transactions on Software Engineering*, 29(8):687–704, 2003.

[17] R. J. Turver and M. Munro. Early impact analysis technique for software maintenance. *Journal of Software Maintenance*, 6(1):35–52, 1994.

[18] K. Wiegers. *Software requirements*. Microsoft Press, 1999.

[19] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers, 2000.