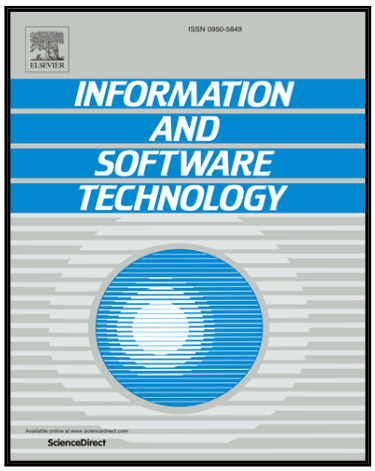

Improving task breakdown comprehensiveness in agile projects with an interaction room

Simon Grapenthin , Steven Poggel , Matthias Book , Volker Gruhn

PII: S0950-5849(15)00131-7
DOI: 10.1016/j.infsof.2015.07.008
Reference: INFSOF 5622

To appear in: *Information and Software Technology*

Received date: 29 November 2014
Revised date: 20 July 2015
Accepted date: 24 July 2015

Please cite this article as: Simon Grapenthin , Steven Poggel , Matthias Book , Volker Gruhn , Improving task breakdown comprehensiveness in agile projects with an interaction room, *Information and Software Technology* (2015), doi: 10.1016/j.infsof.2015.07.008

Improving Task Breakdown Comprehensiveness in Agile Projects with an Interaction Room

Simon Grapenthin [a], Steven Poggel [b], Matthias Book [c,*], Volker Gruhn [a]

[a] paluno – The Ruhr Institute for Software Technology, University of Duisburg-Essen, Gerlingstr. 16, 45127 Essen, Germany
[b] adesso AG, Stockholmer Allee 20, 44269 Dortmund, Germany
[c] Dept. of Computer Science, University of Iceland, Dunhagi 5, 107 Reykjavík, Iceland

* Corresponding author
E-mail addresses: simon.grapenthin@paluno.uni-due.de, steven.poggel@adesso.de, book@hi.is, volker.gruhn@paluno.uni-due.de

ABSTRACT

Context: The planning, estimation and controlling mechanisms of agile process models rely significantly on a fixed set of tasks being established for each sprint. These tasks are created as refinements of product backlog items at the beginning of each sprint. However, a project team's understanding of the backlog items' business implications and dependencies may often not be deep enough to identify all necessary tasks this early, so in addition to the tasks defined in the beginning of the sprint, more necessary tasks might be discovered as the sprint progresses, making any attempt at progress estimation or risk management difficult.

Objective: We strive to enable software teams to achieve a deeper understanding of product backlog items, which should help them to identify a sprint's tasks more reliably and comprehensively, and avoid discovering the need for extra tasks during sprint execution.

Method: We introduced a project team in a medium-sized software development company to the Interaction Room method, which encourages interdisciplinary communication about key system design aspects among all stakeholders. We observed the team's conduct in the sprint planning meetings, and tracked early- vs. late-identified tasks across several sprints.

Results: Before the introduction of our method, the team used to discover on average 26% of a sprint's tasks not at the beginning of the sprint, but later during the course of the sprint. Using the Interaction Room in two separate projects, this ratio dropped to an average of 5% late-discovered tasks.

Conclusion: Our observations from these projects suggest that increased communication among all stakeholders of a project leads to a more reliable identification of the tasks to be performed in a sprint, and that an Interaction Room can provide appropriate guidance to conduct this team communication in a focused and pragmatic way.

Keywords: Scrum, Sprint Planning Meeting, Task Breakdown, Interaction Room

1. Introduction

Scrum is the most popular agile software process model [1]. It structures a project into a series of iterations that last from one to four weeks. Each iteration comprises the steps of iteration planning (in the *sprint planning meeting*), implementing and testing a certain product increment (in a *sprint*), presenting the results to stakeholders (in the *sprint review*) and reflecting on one's work in order to optimize future sprints (in the *sprint retrospective*) [2]. Scrum defines three roles that together form the so-called Scrum team: The *product owner* is responsible for managing the product, defining and prioritizing product requirements. The *Scrum master* manages the process. He or she protects the team from interferences, identifies and eliminates impediments and advocates Scrum within the organization. The *development team* consists of particular roles needed for the development (e.g. developer, tester and architect). The development team is a proper subset of the Scrum team, excluding the Scrum master and the product owner. The development team members manage themselves in terms of how they complete the work to be done in a sprint [3]. The definition of this work occurs in the sprint planning meeting that takes place at the beginning of each sprint. It is divided into two separate meetings: In *sprint planning meeting 1*, the team gathers all information needed to estimate and commit to the implementation of a set of prioritized *backlog items*. A backlog item (or item for short) is a functional requirement defined by the product owner. In *sprint planning meeting 2*, the developers are then supposed to define all *tasks* that need to be performed in order to design, implement and test the backlog items selected for the sprint. Tasks describe the team's planned activities that are necessary in order to fulfill the functional requirements. For each sprint, the development team commits to a limited number of backlog items, based on the estimates of the items' complexity and the team's velocity (i.e. the number of work units that can be handled by the team in a sprint). The velocity is a measurement that becomes more precise the more iterations a team completes, because its members gain experience in estimating the system's complexity and their own performance [4].

Deriving tasks from backlog items is a complex task, however. The team needs to have detailed knowledge about a variety of business and technical aspects of the information system under construction, as well as an understanding of the process and component landscape it is being integrated into, and the data structures it should rely on. Often, slicing tasks out of a backlog item is not straightforward. For example, if the specification of a backlog item is written from a user's