Research plan – Planning product releases in global multi-team agile development projects

Ville T. Heikkilä
School of Science
Aalto University
Helsinki, Finland
ville.t.heikkila@aalto.fi

Advisor: Prof. Casper Lassenius, PhD
School of Science
Aalto University
Helsinki, Finland
casper.lassenius@tkk.fi

Research area: Software engineering

Abstract—Release planning is a challenging part of market-driven product development and a success factor in agile software development projects. Release planning in an environment where multiple globally distributed teams are developing the same product requires careful coordination of responsibilities and technical dependencies. Existing release planning research has concentrated on mathematical model-based big upfront planning, which conflicts with agile methods. The goals of this research are to describe the release planning state-of-art in scientific literature and in real globally distributed agile software development projects, and to describe good practices for efficient release planning in globally distributed agile software development projects.

I. INTRODUCTION

Planning the next product release is recognized to be a critical success factor in agile software development projects [1]. Due to the wicked nature of the problem [2], release planning is also one of the most challenging and important parts of market-driven product development [3], [4]. Immediately after development of a software product starts the schedule starts to deviate from the original release plan. There are many reasons for this schedule variability, but one of the most important is the difficulty of estimating development effort [5], especially when a deeply interconnected product is developed over several months and simultaneously by multiple teams. Typically the intended customers and users of a software product cannot accurately convey what they require from the product before they get some hands-on experience with the product [3]. This requirements variability problem was already identified by [6] (“do it twice”) and later on affected the inception of rapid prototyping [7], [8] and agile software development methods [9], [10], [11]. In a stereotypical waterfall project the variability in schedule does not become apparent until later in the project and the variability in requirements is only revealed after the development project has ended. Iterative and incremental development models such as Rational Unified Process [12] mitigate the schedule variability by dividing the project into relatively short increments, but do not directly address the variability in requirements. Note that requirements variability is not always bad. Sometimes the most uncertain requirements can provide the biggest value [13].

One proposed solution for unwanted requirements variability is to divide the development project into several incremental release projects. Each incremental release project should produce a set of features which are integrated, tested and of sufficient quality. Reinertsen [13] describes several mathematical and practical reasons that explain why reduction of batch size (i.e. incremental releases) is so effective against unwanted variability. According to Reinertsen, reducing batch size reduces variability in the development work flow, accelerates feedback and reduces risk. Reinertsen also states, that large batches inherently decrease motivation and the feeling of urgency, may lead to even larger batches later on, and limit the progress of the batch to the worst (i.e. last) element of the batch. Figure 1 illustrates the amount of uncertainty over a development project with (a) one release or with (b) several incremental releases. \( t_0 \) is the beginning of the project, and \( t_R \) and \( t_R \ldots R_3 \) are releases.

The purpose of this research is to fill the theoretical void that exists between the research domains of release planning and agile software development. As described above, both of these subject areas have been recently identified as important topics in the SE research community.

The research question of this research is “How should software release project planning be performed in agile software development organizations where many globally distributed teams are developing the same product at the same time?”

The main goals of this proposed research are to
1) produce a description of the state of the art in release planning
2) produce a description of the good practices for efficient release planning
in globally distributed multi-team software product development projects which employ agile development methods.

II. RELATED WORK

The majority of publications on software release planning focus on different kinds of mathematical models and simulations designed to create the most valuable, satisfying or risk free release plans when key factors such as resource availability, value of requirements, development effort and
Figure 1. Uncertainty in (a) a single-release project and (b) a multiple-release project.

Risks are known or can at least be estimated somewhat accurately [14]. The model or simulation is then used to generate one or a set of optimal or near-optimal release plan(s). In mathematical release planning research domain such models are called strategic release planning models [15]. To avoid confusion with strategic planning in NPD literature (see e.g. [16]), such models will be referred as mathematical release planning models (or mathematical models) in this research. The emphasis of such models is the mathematical modeling and optimization of the release planning problem. Prioritizing requirements is a central component of any release planning process. Many mathematical release planning models use some variation of a well known prioritization technique such as Analytic Hierarchy Process (AHP) [17] or Cumulative Voting (CV) [18]. Typically such prioritization methods are only applicable when there are relatively few requirements and the requirements can be understood based on a short description [18], although some variants claim to mitigate the scalability problem to some extent [18], [19].

The validity of most of the mathematical release planning models in a large-scale industrial setting is questionable. Svahnberg et al. [14] reviewed 22 mathematical release planning models. Only four of those models had been validated in large-scale industrial use. In addition, the model-driven approach to release planning has proven to be problematic in practice, as many software companies do not have a software development process which could be relied on to record or generate required key factors [20] and the factors might be context dependent and only discovered during release planning [2]. Many models are relatively complex and produce results that give impressions of false exactness and accuracy, which makes it difficult for practitioners to trust the results of such models [2]. Often requirements change so frequently that any long-term plan quickly becomes obsolete [21], and the aforementioned scalability problem\(^1\) inherent to many requirements prioritization methods limits the scalability of the models. Agile software development methods claim to mitigate these issues by not creating detailed long-term plans and by adapting to changing requirements and priorities when needed [22].

Release planning in the agile software development context has only been slightly addressed in scientific publications. Our previous paper described how one organization performed release planning in multi-team agile software development context [23]. The rest of the existing scientific publications [24], [25], [26] propose using a mathematical release planning model in an agile software development context. However, none of the mathematical models were empirically validated in real industrial use.

III. RESEARCH METHOD

The research will include both qualitative and quantitative methods. The primary research methods for the research is case study research [27] and action research [28]. According to [27], case study has distinct advantage over other research methods when "a 'how' or 'why' question is being asked about contemporary set of events over which the investigator has little or no control." Release planning performed in software companies is very much contemporary and a researcher has practically no control over it. However, while researchers will not actively supervise or guide release planning, they will provide feedback and improvements suggestions to the companies. Thus the research must be also regarded as action research and analysis of the research must take into account all that entails [29]. Following classification by [30], this action research can be described as "participant action research". Data collection will be performed using several methods including observations, surveys, artifact analysis and interviews. The different data collection methods combined with the longitudinal and cyclical nature of release planning lends itself to triangulation, following the recommendations of [29]. According to [31] action research interventions in a organization must be based on solid theory which is improved on each cycle of the diagnose-plan-act-evaluate-reflect-process. Case study based theory building literature (e.g. [27], [32]) will provide additional guidance for theory building for building a solid qualitative theory for the research.

IV. RESULTS

This research has resulted in two scientific publications so far. The publications and their contribution to the research goals are described bellow.

1) “Rigorous Support for Flexible Planning of Product Releases - A Stakeholder-Centric Approach and its Initial

\(^1\)For example, the number of comparisons in simple pairwise comparison is \(O(n^2)\).


Publication 1 presents a formalization of the release planning problem, describes a formal release planning process and gives a simple example of release planning in agile software development context (goals 1 and 2). Publication 2 contains a qualitative description of how a Finnish software product company performs agile release planning in a multi-team development project (goal 1).

V. EXPECTED RESULTS AND EVALUATION

Three more publications and a PhD thesis have been planned. The publications are described bellow. The conference articles will be (and have been) evaluated by the international research community in several high quality conferences, the journal article will be evaluated by editors and reviewers, and the doctoral dissertation will be reviewed by experts in the preliminary examination and by the opponent in the public defense.

3) "Lessons Learned from Scaling Agile Release Planning”, The 5th International Conference On Software Engineering and Measurement (ESEM 2011), 2011

4) "Release Planning in Globally Distributed Multi-Project Environment”, Hawaii International Conference on System Sciences (HICSS-45), 2012


6) Doctoral dissertation, spring 2014

Publication 3 describes lessons learned on scaling agile release planning based on a longitudinal multiple case study. Publication 4 describes multiple case study based on studying several companies developing multiple projects simultaneously using the same resource pool. Publication 5 pulls together all previous publications and describes good practices for release planning in large-scale agile software development project. Publication 6 is a doctoral dissertation which contains and summarizes all previous publications.

REFERENCES