

# How does Quality of Formalized Software Processes Affect Adoption?

María Cecilia Bastarrica<sup>1</sup>, Gerardo Matturro<sup>2</sup>, Romain Robbes<sup>1</sup>,  
Luis Silvestre<sup>1</sup>, and René Vidal<sup>3</sup>

<sup>1</sup> Computer Science Department, Universidad de Chile

<sup>2</sup> Departamento de Ingeniería de Software, Universidad ORT Uruguay

<sup>3</sup> Amisoft, Soluciones y Servicios TI

**Abstract.** Defining software processes allows companies to evaluate and improve them enhancing development productivity and product quality, as well as allowing certification or evaluation. Formalizing processes also helps eliminating ambiguity, and enables tool support for evolution and automatic analysis. But these benefits cannot be fully achieved if practitioners do not adopt the process. Some challenges related to adoption have already been identified. In this paper we analyze the influence of the quality of the specified process on its adoption. Adoption is measured in terms of work products built during projects: work products that were not built, those that were built but late during the project, and those that were built in time. We illustrate this analysis by evaluating the adoption of a formalized process in a small Chilean company along five projects. We conclude that certain kinds of errors in process specification may threaten its adoption and thus its potential benefits.

## 1 Introduction

Defining software processes is perceived as beneficial as it allows to count on a pre-established way of working enabling planning, evaluation and improvement. Defined processes also enable companies to obtain an ISO certification or a CMMI evaluation that may provide commercial benefits. Formalizing the defined process also brings the advantages of formal methods: eliminating ambiguity, and being able to automatically analyze the process being able to find errors otherwise hidden [10]. However, if the formalized process is not adopted, all these benefits may not be attained. Process adoption has been identified as a challenge by several authors [12][16], but to the best of our knowledge, it is still not clear how the quality of the software process specification influences adoption.

For the last five years we have been working with small companies in Chile aiding them defining and formalizing their software processes with varying levels of success [19]. We use SPEM 2.0 <sup>1</sup> for the formalization and EPF Composer<sup>2</sup> as

---

<sup>1</sup> SPEM 2.0 - <http://www.omg.org/spec/SPEM/>

<sup>2</sup> Eclipse Process Framework Composer - <http://www.eclipse.org/epf/>

the supporting tool, following the current trend [11]. Some of these companies admit they are not rigorous in following their defined process but they are not aware of the causes. Others say that the resulting process is too detailed and that it is difficult for practitioners to understand and follow. These factors are similar to those reported in the literature [16][17]. However, we have found that companies where all the reported factors affecting adoption were addressed still have trouble in applying the process.

In this paper we focus on the influence that the quality of the software process specification has over its adoption. To this end we have recorded the actual use of the software process in Amisoft, a small Chilean company. This process has been formalized and applied for two years at the moment this research took place. They have achieved an ISO certification and recently a CMMI level 2 evaluation. We registered information about two development and three maintenance projects [18]. For each project, the manager manually tailors the organizational process defining a subset of all the work products that are required. A Gantt chart is built for each project indicating the expected date of construction and modification of each work product. During project execution, expected work products' state is recorder weekly. The process adoption is evaluated in terms of expected work products that are actually built in time, those that are late, and those that never get to be built during the project.

We have built AVISPA (Analysis and Visualization for Software Process Assessment) [10], a tool that builds blueprints and highlights error patterns for a given process model. Error patterns are identified with process elements that are graphically abnormally different from the remaining elements [13]. Counting on this tool, the process engineer only needs to analyze highlighted elements, demanding little experience and also little previous knowledge for effective process model analysis, adding usability as well. Some of these errors are the existence of overloaded roles, too demanded work products and waste work products, among others. These situations indicate possible errors in the process design or specification. We use some of these patterns and create new ones in order to try to identify characteristics of the process specification that may affect its adoption. Among our hypotheses about the possible factors related to the specified process quality that may threaten process adoption we find:

1. no role is responsible for the work product,
2. two or more roles are responsible for the work product,
3. too many roles are assigned to the work product,
4. the responsible role is overloaded,
5. no templates are available for the work product,
6. the work product is potentially waste, i.e., nobody needs it.

The rest of the paper is structured as follows. Section 2 presents background concepts about software process adoption and several factors affecting it. The hypotheses about the issues in process quality that may affect adoption and their application to the company's process are discussed in Sect. 3. The empirical study conducted in the company is described in Sect. 4. Section 5 reports the data analysis and the validation of the hypotheses. A survey conducted to check

our hypotheses with the company's employees is described in Sect. 6, the threats to the validity are described in Sect. 7 and some conclusion are stated in Sect. 8.

## 2 Software Process Adoption

Several factors can influence the successful deployment and adoption of new process models. At the individual level, Chroust considers that process models and their support in the form of a software engineering environment, have a high pervasiveness and impact on the software development process and the people working with it [2]. To this author, an individual's views, perceptions and motivations to accept and enact software process models outweigh all other factors with respect to quality, cost, and time, and are key factors for success. Among the human or "soft" factors, he mentions the need for learning the new process and the additional qualifications required, the fear of added complexity, the loss of flexibility and creativity, and the amount of new documentation that must be created and kept up to date, in detriment to "productive work". Even though we have witnessed the influence of these factors on adoption, we focus on the influence that process specification quality has on its adoption. We address some of these issues in the employees' interviews in Sect. 6. At the organizational level, Rifkin considers that the difference between success and failure in the adoption of software process innovations is related to upper-management commitment and sponsorship, the ability or persuasiveness of change agents, the divisibility of the innovation, how disruptive the innovation is, and whether the change is planned and managed [17]. This issue is actually completely necessary in practice to enable any kind of success in adopting formalized software processes. The company that illustrates this research has a highly qualified manager that encourages people involved in development to strictly follow the defined process.

Other factors that influence process adoption in small companies, had also been identified in [20]: notation expressiveness and understandability, supporting tools usability and availability, and the level of interoperability of these tools with others used by the company. On the other hand, O'Connor and Coleman have studied the barriers to the adoption of software process models that are based on industry "best practices", such as CMM/CMMI and ISO 9000. The main factors they have found are Documentation (overhead which can delay development activity and whose merits are difficult to convey to engineers), Bureaucracy (time and resources which the managers believe are required to manage and apply the software process), and the perception of limiting Creativity and Flexibility [14].

As stated by Persee [15], the trick to the success will not come from merely building a good process; it will come from using it: using it over time, refining it, making it better and better, and allowing it to become a permanent part of the organization's business approach. We try to isolate in this paper some of the factors that may promote the adoption of a good process. The introduction of a new process into an organization can be seen as a particular case of an innovation in the way software development activities have to be done in the future. Hammed, Counsell and Swift define innovation as an idea, a product, a process

or a technology that is new to the adopting unit, and also define adoption of an innovation as the process that results in the introduction and use of a product, process, or practice that is new to the adopting organization [8]. When the innovation in question involves a change to software development practices, there are particular difficulties in encouraging individual developers to adopt and sustain the use of disciplined, repeatable processes [7]. In particular, if a methodology is not regarded as useful by developers, its prospects for successful deployment may be severely undermined. In other words, to the extent a methodology is not useful, that is, it does not enable developers to be more productive and achieve higher levels of performance in their job, they are not likely to use it in a sustained manner, even if it is mandated [16]. We establish some hypotheses about the causes that make motivated developers not to use the process and we check them against historical data of five projects in the same company.

One theoretical model widely used for addressing the technology adoption process is the technology acceptance model (TAM) [3], which is based on the relationships among perceived attributes of a technology, attitudes toward a technology, and actual usage. This model employs the constructs perceived usefulness of a technology and perceived ease-of-use of it as main determinants for explaining adoption. We use some of these concepts for building the survey questions described in Sect. 6. In [4], they analyze the factors that affect software developers' acceptance and utilization of electronic process guides. The results show that perceived usefulness is a strong and highly significant determinant of current system usage and future use intentions. If a process is represented in electronic or paper-based form, it is not regarded as useful and it will not be fully adopted by developers. As stated by Heijstek and van Vliet, most of the times new processes are adopted to some extent, but not to the extent needed to achieve any real benefit [9]. Partial adoption can come in the form of just performing some of the activities prescribed by the model, not performing the activities in the way they are defined, and/or when not all of the defined work products are elaborated and used during process execution.

This leads to define adoption of a software process as the fact that the process model is followed to the extent that ensures process compliance, that is, when performing the process, developers actually comply with process requirements in order to ensure process success and delivery of desired outcomes. In defining a process model one can take two different but complementary approaches. One is the "activity approach" that focuses on tasks or activities that relate to work to be done, and the other one is the "work product approach", that focuses on work products that relate to artifacts to be built. According to Gonzalez-Perez and Henderson-Sellers, a methodology that focuses on expressing its work products rather than its process is arguably more people-oriented, being better at dynamically reorganizing the work to be done and opportunistically exploiting unforeseen circumstances [5]. As noted by Goodman, this last perspective can support work product "states" for work product promotion/demotion, and assumes that work product production and activity execution are synonymous [6]. From this second perspective, process adoption can be reflected on the work

products actually elaborated and used during process execution. This is precisely the approach taken in this study, in order to analyze process adoption in the target company.

### 3 Process Model Quality Analysis

We say that a software process is not fully adopted if there are work products that, being necessary for a project, have either not been developed, or have been late during the project execution. We state a series of hypotheses that may explain why a software process could have adoption difficulties, and analyze the APF process from the point of view of these hypotheses. The hypotheses are derived from our hands-on experience with the software companies collaborating with the ADAPTE project<sup>3</sup>, documented in our previous work [1,10], where we analyzed the software processes of Amisoft, BBR Engineering, and DTS (hypotheses 1, 4, and 6). Additional hypotheses were formulated and refined based on the additional analysis that we carried on for this paper (hypotheses 2, 3, and 5).

(1) *No role is responsible for the work product.* EPF Composer allows to assign a role responsible for each work product. This role should be in charge of assuring that the work product is appropriately built and modified. If there is nobody in charge of the timely construction of the work product, it is likely that nobody would take responsibility for it, so not assigning a responsible role for a work product may cause it not to be built in time. AVISPA allows identifying work products with no responsible role assigned. Analyzing APF with this pattern we found that only three work products had no responsible role.

(2) *Two or more roles are responsible for the work product.* On the other hand, there may be two or more roles responsible for a certain work product. In this case, although the process specification makes sure at least one role will take care of the work product, this situation of shared responsibility may prevent all of them to assume it. For computing this situation we extend AVISPA so that it colors those elements with two or more responsible roles. We found five work products with two or more responsible roles in APF.

(3) *Too many roles are assigned to a work product.* There are certain work products that are used and/or modified by more than one role. This is not necessarily a problem in itself, but if there are too many roles that interact with a work product coordination may become an issue for finishing appropriately and in time. We assumed that up to three roles would be fine, but having four or more roles modifying a work product could be a source of problems. We extended AVISPA so that those work products with four or more roles modifying it are colored, and we found several of these elements.

(4) *Overloaded role is responsible for the work product.* Overloaded roles are those assigned to too many work products, either as responsible or modifying them.

---

<sup>3</sup> [http://www.adapte.cl/?page\\_id=36](http://www.adapte.cl/?page_id=36)

If an overloaded role is in charge of a work product, he/she may have other priority tasks to do. AVISPA provides an error pattern for identifying overloaded roles. The Project Manager is by far the role involved in more work products in APF, so his work products are candidates to be delayed.

(5) *No template is available for the work product.* EPF Composer allows to associate a template to each work product. Counting on a predesigned template makes it easier to build the work product in time so we expect that those elements with no template are more likely to be delayed. We extended AVISPA creating a new error pattern that colors those work products with no template associated. We found only a few work products in APF with no template.

(6) *Potentially waste work product.* If a work product needs to be used as an input for certain task but the specified process does not state it, this task may not be correctly executed. This situation is not always easily detected. However, if a work product is neither marked as deliverable nor it is specified as input of a task, it is potentially waste. Waste work products are those that are neither needed for any task in the process nor deliverables. The AVISPA tool is able to identify potentially waste work products. APF presents several work products that are potential waste, at least in the formalized process; some of them may be underspecifications (i.e., input for some task or deliverables), or actually waste work products.

## 4 Empirical Study

### 4.1 Case Study

Our case study was conducted at Amisoft, a software services company based in Santiago, Chile, that builds and maintains custom-order software. Amisoft employs 43 people, and has on average 2 development contracts a year and 7 permanent maintenance contracts. In the last years, Amisoft transitioned from an informal software development process, to a formally specified process –APF– based on the Rational Unified Process (RUP). Further, in order to know if the process is a net benefit for the company and if employees actually follow it, Amisoft keeps track of the artifacts that are produced during projects. A full-time metric analyst is in charge of tracking the adoption to the process.

We study 2 development and 3 maintenance projects developed at Amisoft. Table 1 contains summary statistics of the projects. SITMIX, SITLA, SITCO and SITCORTE are systems specifically developed for the Chilean Judiciary. They allow to automate manual processes and to eliminate paperwork. The main functionality of these systems is a configurable workflow integrated with a document management system. Currently Amisoft is in charge of the maintenance of these systems for its customers; that is why only SITMIX is a development project for our case study. AMILEX is a product developed by Amisoft that encapsulates the knowledge acquired in paperless management workflow systems. Since the process is tailored for every project, the work products created during each project vary. In particular, there are (unsurprisingly) significant differences

**Table 1.** Description of Projects Analyzed in the Empirical Study

Project	Type	Team	Months	Description
<b>SITMIX</b>	Dev.	1 project manager + 5 developers	12	Mixed courts information system
<b>AMILEX</b>	Dev.	1 project manager + 3 developers + 0.5 architect	7	Parameterizable workflow and digital folder
<b>SITLA</b>	Maint.	0.5 project manager + 3 developers	6	Labor Courts information system support
<b>SITCO</b>	Maint.	0.5 project manager + 3 developers	6	Collection, Labor and Social Courts information system
<b>SITCORTE</b>	Maint.	0.5 project manager + 5 developers	6	Courts of Appeal information system

between the processes followed by development and maintenance projects; in addition, individual processes vary, particularly among development projects.

## 4.2 Methodology

In this study, we perform a two-step investigation: we first analyze the process adoption spreadsheet of each project to confirm our hypotheses, and then we perform a follow-up employee questionnaire of the hypotheses.

*Process adoption spreadsheet.* The process adoption spreadsheet is maintained by the metrics analyst, and it tracks each work product weekly. Each work product that is expected to be delivered before that date is checked in order to know if it is present. The work product is then marked, for that week, as either delivered or late. At the end of the project, the weekly information is consolidated in order to define 3 possible statuses for a given work product:

- **Delivered on time (OT).** The work product was present in the project, and each time its delivery was expected, it was indeed present.
- **Delayed (D).** At least once during the project, the work product suffered a delay of at least one week.
- **Not delivered (ND).** At the end of the project, the work product was still not delivered.

Based on the data that we have for each work product of each project, we look for evidence towards the validation or invalidation of each of our hypotheses.

*Employee survey.* In order to follow up on the first analysis, we also conducted a survey of the project managers of Amisoft, specifically designed to provide further evidence for or against our hypotheses. Project managers are all Computer Science Engineers with more than three years experience. They are in charge of scheduling activities, controlling that milestones are met, and making sure that team members apply the company's software process. They are also in charge of requirements analysis for their projects.

**Table 2.** General work product statistics

Project Id	Work products	OT	D	ND
SITMIX	30	18	10	2
AMILEX	37	22	13	2
SITLA	19	7	12	0
SITCO	34	27	3	1
SITCORTE	34	28	3	0

## 5 Process Adoption Spreadsheet

Table 2 presents the general results of our study. Each row presents a project, and for each project: (1) the total number of work products in the tailored process; (2) the number of work products that were delivered in time; (3) the number of work products that were delayed; and (4) the number of work products that were not delivered. From this table, we can see that a minority of work products are not delivered (which is reassuring for Amisoft). For development projects, a third of the work products experience delays or are not delivered. For maintenance projects, there are fewer delays, with more than 80% of the work products being delivered in time, which is within Amisoft’s goals. Table 3 presents a summary of the work products, whether they comply with our hypotheses, and, for each project, whether they were delivered on time, or experienced issues.

(1) *No role is responsible for the work product.* We found weak evidence for this claim. Only three work products fit the pattern (MUS, ICF, DDD). Of those, DDD is only present in the process of project AMILEX; ICF has been delivered in time each time, and MUS has not been delivered for project SITCO. In all cases, MUS, ICF, and DDD are subsumed by the hypothesis that they may be waste as well; as such, the evidence is overall inconclusive.

(2) *Two or more roles are responsible for the work product.* We again find that a low number of work products suffer from this issue. There are 5 overall: TEC, CGP, SRQ, ARR, and PMD. Of those, PMD was the only one that did not encounter delays of any kind. On the other hand, TEC and CGP were late in both development projects, while SRQ was not performed in SITMIX, and ARR was delayed in AMILEX. Of note, TEC was late in both SITMIX and AMILEX, but it was performed in time for all three maintenance projects; these 3 projects did not include CGP, ARR, or SRQ in their processes. Overall, it seems that there is reasonable initial support for the hypothesis, even though the number of work products concerned is small, and 3 out of 5 of these work products were not present in maintenance projects.

(3) *Too many roles are assigned to the work product.* We consider the threshold of too many roles to be 4. Using that threshold, we find that 11 work products have too many roles involved for their proper implementation (LAA, CGP, CGI, ARR, MRR, PCC, PPP, PVP, PRP, PMD, and ACE). Of those, 9 feature a delay on at least one project, which yields strong evidence towards this factor

**Table 3.** Work product behavior for each hypothesis: H1 - no responsible role, H2 - two or more responsible roles, H3 - too many interacting roles, H4 - assigned to an overloaded role, H5 - no template available, H6 - defined as waste. For the projects the meaning is: **X** - not delivered, **!** - delayed, **✓** - in time, **--** - not in process

Work product	Hypotheses						Statuses				
	H1	H2	H3	H4	H5	H6	SITMIX	AMILEX	SITLA	SITCO	SITCORTE
LNE	N	N	N	Y	N	Y	X	X	-	-	-
TEC	N	Y	N	Y	N	N	!	!	✓	✓	✓
LAA	N	N	Y	Y	N	Y	!	!	✓	✓	✓
PEE	N	N	N	Y	N	N	!	!	!	✓	✓
CGP	N	Y	Y	Y	N	N	!	!	-	-	-
CGI	N	N	Y	Y	N	N	!	!	✓	✓	✓
SRQ	N	Y	N	Y	N	N	X	✓	-	-	-
ARR	N	Y	Y	Y	N	N	X	✓	-	-	-
MRR	N	N	Y	Y	N	N	!	✓	!	✓	✓
PAC	N	N	N	N	N	N	!	✓	✓	✓	✓
PRH	N	N	N	Y	N	N	!	✓	✓	✓	✓
PRV	N	N	N	Y	N	N	!	✓	✓	✓	✓
ESC	N	N	N	Y	N	Y	✓	X	✓	✓	✓
PEA	N	N	N	Y	N	Y	✓	!	✓	✓	✓
PCC	N	N	Y	Y	N	Y	✓	!	-	-	-
PPR	N	N	N	Y	N	Y	✓	!	!	!	!
PPP	N	N	Y	Y	N	N	✓	!	✓	✓	✓
LRR	N	N	N	Y	N	N	✓	!	✓	✓	✓
PVP	N	N	Y	Y	N	N	✓	!	!	!	!
CPU	N	N	N	Y	N	N	-	!	✓	✓	✓
PRP	N	N	Y	Y	N	Y	-	!	!	!	!
VPP	N	N	N	Y	N	N	✓	✓	-	-	-
PLP	N	N	N	Y	N	N	✓	✓	-	-	-
PMD	N	Y	Y	Y	Y	N	-	✓	✓	✓	✓
PAP	N	N	N	N	N	N	✓	✓	✓	✓	✓
MCU	N	N	N	Y	N	N	-	✓	-	-	-
MDD	N	N	N	N	N	N	✓	✓	✓	✓	✓
MUS	Y	N	N	N	Y	Y	✓	✓	✓	X	✓
ICF	Y	N	N	N	Y	Y	✓	✓	✓	-	✓
PIR	N	N	N	N	N	N	-	✓	✓	✓	✓
GLS	N	N	N	Y	N	Y	✓	✓	✓	✓	✓
ESS	N	N	N	N	N	N	✓	✓	✓	✓	✓
ERS	N	N	N	Y	N	N	✓	-	✓	✓	✓
DDD	Y	N	N	N	Y	Y	-	✓	-	-	-
DDR	N	N	N	N	Y	Y	-	✓	✓	✓	✓
SAD	N	N	N	N	Y	N	✓	✓	-	-	-
ACE	N	N	Y	Y	N	Y	-	✓	✓	✓	✓
Total	3	5	11	15	5	13	-	-	-	-	-

being a possible cause for delays. As said above, the higher the number of roles, the more probable these coordination issues arise. We also note that development projects are more concerned with this issue (9 work products) than maintenance projects (only 3 work products are affected).

(4) *Overloaded role is responsible for the work product.* The most overloaded role by far is the project manager; therefore, we only investigate this particular role. Our analysis brings strong evidence that if a role is overloaded, it is a factor in work products that are delayed or not delivered: of the 17 work products the project manager is involved in, 15 experienced delays. Clearly, the project manager has many things on his plate, and a lot of them can pass through. We note that of these 15, only 2 concern maintenance projects (15 concern development projects), highlighting again the asymmetry between both kinds of

projects. It seems that if changes to the process could delegate some of these tasks to other roles, this would significantly help process adoption all around.

(5) *No templates are available for the work product.* We do not find support for this hypothesis. The presence/absence of a template seems uncorrelated with the adoption of the process. If anything, it appears to be negatively correlated: of the six work products that do not have a template (PMD, MUS, ICF, DDD, DDR, SAD), only one, MUS, encountered issues, in only one project.

(6) *Potentially waste work product.* Waste is defined as a work product that is generated, but that is neither deliverable nor used in any other task. This hypothesis seems to be a good factor for explanation of adoption as well, although the reasoning is not obvious at first glance. There are 13 work products marked as “waste”. Of those, only 8 suffer any kind of delay, which does not provide a very strong support for waste being an issue. However, examining the work products marked as waste, we find that several of them were victim of underspecification. More precisely, several work products marked as “waste” were actually deliverables to the client, and as such employees are very aware of their importance. The waste work products that are actually client deliverable are: ACE, DDR, GLS, MUS, and ICF. Of these, only MUS was not delivered; all other work products were delivered, on time. Removing these 5 work products from the “waste” category yields a different picture: of the 8 work products that remain, only DDD does not suffer any delay, which makes “real waste” work products a strong predictor of adoption to the process. In short, perceived utility seems like an important factor. Of note, taking into account the fact that deliverables to the clients are seen as important further weakens the evidence towards the no-responsible role hypothesis, as two of the three work products without a responsible role are deliverables to the client.

## 6 Survey Results

Since we have several competing hypotheses, we attempted to shed more light by getting feedback from the project managers of Amisoft. Specifically, we sent a questionnaire via email to all the project managers, asking them whether they recalled having experienced delays in delivering work products according to our hypotheses. The questionnaire contained one question per hypothesis, plus a last question asking for other causes for the delays. All responses were free-form text, that we analyzed and summarize here. Five project managers replied to our questionnaire, although the fact that it was done by email, and that the managers are busy, had the consequence that some responses were quite succinct and as such not always easy to interpret. Of note, 4 of the 5 managers that responded were from projects we investigated in the first step, while the last one was not.

*No role responsible for the work product.* The project managers were unanimous on this hypothesis: they do not recall any instance of this fact being the cause of delays, going so far as saying that all work products have assigned roles. Given that a small minority of work products were found to have no responsible role, this confirms that our first hypothesis has very weak support in practice: even

if there is (in rare instances) a work product that is not assigned to any role formally in the process, someone in practice is responsible for it.

*Two or more roles responsible for the work product.* We find limited additional support for this hypothesis. In particular three of the managers mentioned that the responsibilities for some work products are sometimes unclear, especially for the testing plan (PPR), resource allocation (ARR), and the requirement specification (ERS). However the additional evidence is limited since both PPR and ERS do not have two responsible roles—only several involved roles—in the process definition, although in practice the distinction between responsible and involved role might not be as clear-cut as it is in the process specification.

*Too many roles assigned to a work product.* One product manager replied saying that due to the size of his team, having too many people involved was not a problem. Given that that team was SITMIX, the largest one in the projects we analyzed, we can extrapolate that other teams did not run into that problem either. As such, it seems that this may be a valid hypothesis, but not in the particular circumstances at Amisoft, where teams are small. On the other hand, another manager mentions that the work product ERS has too many roles, and that it is a problematic situation. Another manager mentions a work product that involves an architect, a developer, and the project manager, but does not precise which. Overall, managers have conflicting opinions about this hypothesis, preventing us from reaching a clear conclusion.

*Overloaded roles are responsible for the work product.* The evidence is much clearer for this hypothesis. All managers agree that the project manager is overloaded. Two managers mention that this is an explicit risk to the project, that is taken into account in order to mitigate it (by dividing up tasks, and delegating them to other people). One manager is also aware of this strategy to reduce his load, but mentions he is unable to delegate tasks, since his team is too small. Finally, another manager mentions the quality assurance (QA) roles as also being affected by overloading, although he is the only one.

*No templates are available for the work product.* For this hypothesis, all managers find that the templates are present, and are adequate, except for specific work products, where they could be improved (with TEC being explicitly mentioned once). As such, the evidence gathered by the questionnaire leads us to reject this hypothesis, at least for the case of projects at Amisoft.

*Potentially waste work product.* We find convincing evidence that there is perceived waste in the process, coming from three of the project managers. For some work products, it is unclear who will use them once they are produced, or why they are important. A few work products are mentioned by name. These include PRV, ERS, PLP, and PCC, and (less precisely) activities related to verification and validation, and activities related to process monitoring (i.e., metric collection regarding the process itself). From this, we can conclude that waste, as perceived by the managers, is an issue that is present in the process.

*Other reasons.* Finally, there are several other reasons for delays that are mentioned in the free entry space that we let for the project managers to fill. There are two main reasons:

- Dependencies between tasks is a problem referenced by four of the five interviewed managers. By dependencies between tasks, we mean the situation that arises when a task is delayed because of waiting upon the completion of another task. In particular, three managers mention the Quality Assurance area as being a bottleneck in that regard.
- External factors, such as changes in personnel, and changes to requirements made by clients are mentioned as source of delays, that propagate to other tasks. However these factors cannot be attributed to the process itself.

## 7 Discussion

### 7.1 Aggregating the Evidence

Combining the evidence we gathered in the two steps of our investigation, we can order our hypothesis from the least likely to the most likely to incur delays in delivering work products.

#### *Unlikely hypotheses*

- *No role responsible for the work product:* we found very few work products which did not have a responsible role specified, and a low correlation with delays or work products undelivered. Further, all project managers stated that all work products are in the responsibility of someone.
- *No templates are available for the work product:* we found a similar situation as with the previous hypothesis. There are few work products matching the hypothesis, a low correlation, and projects managers concur.

#### *Moderately likely hypotheses*

- *Two or more roles responsible for the work product:* We found that there are few work products matching with the hypothesis, but that they have a high correlation with problematic work products. On the other hand, other hypotheses seem to have more explaining power, and project managers gave little additional support in the questionnaire.
- *Too many roles assigned to a work product:* This hypothesis has a good correlation with problematic work products. However, project managers have conflicting opinions about it, with at least one manager saying that Amisoft's teams are too small for it to be an issue.

#### *Likely hypotheses*

- *Potentially waste work product:* at first, we found a moderate correlation between potential waste and problematic work products. However, a detailed inspection reveals that some of the "waste" is not really waste, as it concerns work products directly relevant to the client (deliverables), hence which are very likely to be done anyways. With this new knowledge in hand, we find that waste

is a much better explanation. This is further confirmed by the managers, of whom most agree that there is waste in the process, conforing this hypothesis.

- *Overloaded roles are responsible for the work product:* finally, this hypothesis has the strongest support. It has a strong correlation with problematic work products, and managers strongly echo this in their comments.

## 7.2 Difference between Development and Maintenance Projects

We observe that maintenance projects have less problematic work products than development ones. A follow-up with the company's CEO gave us some possible reasons, for whom the most important is the type of contract for each type of project, and the impact this has on the process. Maintenance projects usually have a monthly contract, while development projects have contracts based on functionality milestones. As such, there is more pressure in development projects to produce deliverable work products and functionality fast, leading to a lower priority towards other artifacts in the process, which are either for internal use, or that are perceived to have a lower added value (such as verification and validation activities).

## 7.3 Additional Hypotheses

Beyond the hypotheses checked, there are others not investigated yet.

- *Synchronization issues.* Managers reported that some tasks were delayed because they were waiting on other tasks to be finished, that behave as bottlenecks (e.g., QA activities). Since managers are keenly aware of this issue, it seems to be a salient one.

- *Early vs late tasks.* Related to synchronization issues, the fact that a task is late in the project or in an iteration could lead to it being more likely to be delayed, as the delays tend to accumulate with time.

- *Verification and validation.* Managers and the company's CEO mentioned on several occasions that tasks from the Verification and Validation software engineering discipline frequently were the most problematic. A quick glance at the problematic work products reveals that the most problematic ones (the ones delayed in 4 out of 5 projects), belong to the verification and validation area: PPR, PVP, and PRP. As such, exploring this in more details would be worthwhile.

- *Internal vs external tasks.* Managers often put more emphasis on work products that are more relevant to the client, as progress is more easily seen that way. We have seen hints of this being a factor when we investigated waste, but this phenomenon deserves a fuller investigation.

## 7.4 Threats to Validity

As with any empirical study, this work is subject to several threats to validity, that we summarize in Table 4.

**Table 4.** Threats to validity of the study

<b>Threat</b>	<b>Description</b>
Single case study.	We only analyzed 5 projects from a single company. Different companies will have different characteristics, in terms of size of the company and of the teams working there, different market niches, different cultures, and different software engineering practices. Further, each project has a host of specificities that make it unique, such as the client, the composition of the team, etc. Our study needs to be replicated in a variety of distinct contexts to see whether the findings hold in general, or are specific to this case.
No qualitative analysis of the work products	So far, the process monitoring at Amisoft considers that a work product is delivered only by its presence or absence in the system. There is no inspection of the quality of the work product itself. Some of the work products may be present in the system, but in an incorrect or incomplete state. A manual analysis of the work products would be necessary in order to know if this is the case.
Effect of process tailoring	At Amisoft, each project has a specific process, tailored from the generic process. As such, the set of artifacts to be delivered for each project may vary. The fact that some artifacts were not delivered because they were not in the process to start with makes it harder to compare projects with one another.
Project manager questionnaires	Our questionnaires also have several potential issues. We only sent the questionnaire to project managers, as it would have been prohibitive to send it to all employees at Amisoft, in terms of the effort involved both at Amisoft to fill it (for that reason we also refrained from asking for subsequent clarifications), and on our side to aggregate the information. We chose to send the questionnaire to project managers since they have a good view of their team members, but this choice also incurs bias in the responses.

## 8 Conclusion

In this paper we analyzed the impact that the quality of formalized software processes has on their adoption. We stated six hypotheses about the reasons that may cause work products to be delayed or never built during project execution. The process itself was analyzed using AVISPA, a graphical tool that highlights potentially problematic elements in the process.

The formalized software process adoption was analyzed in two different dimensions. First an empirical study about actual use of the process in practice was conducted in two development and three maintenance projects in the same company. In this study each work product that was either late or not developed in each project was recorded. And then a questionnaire was applied to all five project managers about their personal perception about the stated hypotheses.

We could realize that the process was well specified for defining one and only one responsible for each work product, and there is also a template defined for each work product in the process model. So these issues were not supposed to be the causes for not following the process as expected, at least in this company. Both, the empirical study and the questionnaire corroborated these facts. We found some evidence that work products where four or more roles interact or those where there are more than one role responsible for them are somewhat likely to be delayed. Finally, we found strong evidence that work products that are specified as waste and those that are responsible of an overloaded role are almost always late. In general, all six hypotheses seemed reasonable, but empirical analysis resulted that they were not true in all cases.

## References

1. Bastarrica, M.C., Hurtado, J.A., Bergel, A.: Toward Lean Development in Formally Specified Software Processes. In: 18th EuroSPI 2011, Denmark (June 2011)
2. Chroust, G.: Soft Factors impeding the Adoption of Process Models. In: Proceedings of the 28th EUROMICRO 2002, Dortmund, pp. 388–394 (September 2002)
3. Davis, F.D., Bagozzi, R.P., Warshaw, P.R.: User acceptance of computer technology: A comparison of two theoretical models. *Management Science* 35(8), 982–1003 (1989)
4. Dyba, T., Moe, N.B., Mikkelsen, E.M.: An Empirical Investigation on Factors Affecting Software Developer Acceptance and Utilization of Electronic Process Guides. In: METRICS 2004, Chicago (September 2004)
5. Gonzalez-Perez, C., Henderson-Sellers, B.: A work product pool approach to methodology specification and enactment. *Journal of Systems and Software* 81(8), 1288–1305 (2008)
6. Goodman, F.A.: *Defining and deploying software processes*. Auerbach Publications, Boca Raton (2005)
7. Green, G.C., Collins, R.W., Hevner, A.R.: Perceived control and the discussion of software process innovations. *Journal of High Technology Management Research* 15(1), 123–144 (2004)
8. Hameed, M.A., Counsell, S., Swift, S.: A conceptual model for the process of IT innovation adoption in organizations. *Journal of Engineering and Technology Management* 29, 358–390 (2012)
9. Heijstek, A., van Vliet, H.: Less is More in Software Process Improvement. In: M. Biro and R. Messnarz (eds.) EuroSPI 2005, pp. 6.1–6.12 (2005)
10. Hurtado, J.A., Bastarrica, M.C., Bergel, A.: AVISPA: A Tool for Analyzing Software Process Models. *Journal of Software Evolution and Process* (2013), <http://onlinelibrary.wiley.com/doi/10.1002/smr.1578/abstract>
11. Kuhrmann, M., Fernandez, D.M., Steenweg, R.: Systematic software process development: where do we stand today? In: ICSSP 2013, pp. 166–170 (May 2013)
12. Münch, J., Armbrust, O., Kowalczyk, M., Soto, M.: *Software Process Definition and Management*. Springer (2012)
13. Nierstrasz, O., Ducasse, S., Demeyer, S.: *Object-Oriented Reengineering Patterns*. Square Bracket Associates (2009)
14. O'Connor, R., Coleman, G.: An investigation of barriers to the adoption of software process best practices models. In: Proceedings of the 18th Australasian Conference on Information Systems, Toowoomba, Australia (December 2007)
15. Persee, J.: *Process Improvement Essentials*. O'Reilly, Sebastopol (2006)
16. Riemenschneider, C.K., Hardgrave, B.C., Davis, F.D.: Explaining Software Developer Acceptance of Methodologies: A Comparison of Five Theoretical Models. *IEEE Transactions on Software Engineering* 28(12), 1135–1145 (2002)
17. Rifkin, S.: Why software process innovations are not adopted. *IEEE Software* 18(4), 110–112 (2001)
18. Robbes, R., Vidal, R., Bastarrica, M.C.: Are Software Analytics Efforts Worthwhile for Small Companies? The Case of Amisoft. *IEEE Software* 30(5), 46–53 (2013)
19. Ruiz, P., Quispe, A., Bastarrica, M.C., Hurtado, J.A.: Formalizing the Software Process in Small Companies. In: 8CCC, Colombia (August 2013)
20. Simmonds, J., Bastarrica, M.C., Silvestre, L., Quispe, A.: Variability in Software Process Models: Requirements for Adoption in Industrial Settings. In: PLEASE 2013, San Francisco, USA (May 2013)