

Exploration of Technical Debt in Start-ups

Supplemental material

Eriks Klotins*
Blekinge Institute of Technology
Karlskrona, Sweden
eriks.klotins@bth.se

Michael Unterkalmsteiner
Blekinge Institute of Technology
Karlskrona, Sweden

Panagiota Chatzipetrou
Blekinge Institute of Technology
Karlskrona, Sweden

Tony Gorschek
Blekinge Institute of Technology
Karlskrona, Sweden

Rafael Prikladnicki
Pontifical Catholic University of Rio
Grande do Sul
Porto Alegre, Brazil

Nirnaya Tripathi
University of Oulu
Oulu, Finland

Leandro Bento Pompermaier
Pontifical Catholic University of Rio
Grande do Sul
Porto Alegre, Brazil

KEYWORDS

Software start-ups, technical debt

ACM Reference Format:

Eriks Klotins, Michael Unterkalmsteiner, Panagiota Chatzipetrou, Tony Gorschek, Rafael Prikladnicki, Nirnaya Tripathi, and Leandro Bento Pompermaier. 2018. Exploration of Technical Debt in Start-ups: Supplemental material. In *Proceedings of International Conference of Software Engineering (ICSE'18)*. ACM, New York, NY, USA, Article 4, 5 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

In this document we provide a complete list of questions and answer options that were used to estimate aspects of technical debt in start-ups.

*Corresponding author

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ICSE'18, May 2018, Gothenburg, Sweden

© 2017 Copyright held by the owner/author(s).

ACM ISBN 123-4567-24-567/08/06...\$15.00

https://doi.org/10.475/123_4

Table 1: Questions capturing start-up and respondent demographics

#	Question	Answer options
1	What is the current state of the start-up?	(1) Operational, the team continues working on the product/service (2) Paused, the team has stopped working however there is an intention to resume (3) Acquired by another company (4) Closed down, the team is disbanded or works on something else
2	What is the last known state of the product/service created in the start-up?	(1) A product prototype is developed and has not yet been released to market. (2) Product was released to the market and is actively developed further with customer input. (3) Product is rather stable, the focus is on gaining customer base. (4) Product is stable, market size, share and growth rate are established. Focus is set on launching new variations of the product.
3	To what extent is the product/service tailored to individual customers?	(1) The start-up does not offer per-customer customization (2) Occasionally, the product is tailored to suit needs of an individual customer (3) More often than not, the product is tailored to suit needs of an individual customer (4) Each product or service instance is tailored for an individual customer (5) I do not know
4	Are there any other products developed or services provided at the same time as the primary product/service?	(1) No (2) Yes (3) I do not know
5	How many people are in the start-up team?	(1) 1 - 3 (2) 4 - 8 (3) 9 - 12 (4) 13 - 15 (5) 16 - 20 (6) More than 20 (7) I do not know
6	How many people in the start-up work primarily on software development?	(1) 1 - 3 (2) 4 - 8 (3) 9 - 12 (4) 13 - 15 (5) 16 - 20 (6) More than 20 (7) I do not know

Table 2: Questions capturing start-up and respondent demographics (continued)

#	Question	Answer options
7	What level of software engineering skills does the team have?	(1) Inadequate for the size and complexity of the product/service (2) Somewhat adequate but with many gaps (3) Adequate with minor gaps (4) Adequate and sufficient to develop the product/service without any difficulties (5) I do not know
8	What level of domain knowledge does the team have?	(1) Inadequate for the size and complexity of the product/service (2) Somewhat adequate but with many gaps (3) Adequate with minor gaps (4) Adequate and sufficient to develop the product/service without any difficulties (5) I do not know
9	What experience with similar products/services does the team have?	(1) Inadequate for the size and complexity of the product/service (2) Somewhat adequate but with many gaps (3) Adequate with minor gaps (4) Adequate and sufficient to develop the product/service without any difficulties (5) I do not know
10	When did the team start working on the product?	Date input
11	How much practical experience have you had in your area of expertise at the time when you joined the start-up?	(1) Less than 6 months (2) 6 - 12 months (3) 1 - 3 years (4) 4 - 6 years (5) 7 - 9 years (6) 10 or more years
12	How much experience have you had working in/with start-ups at the time when you joined the start-up?	(1) Less than 6 months (2) 6 - 12 months (3) 1 - 3 years (4) 4 - 6 years (5) 7 - 9 years (6) 10 or more years
13	Which option most precisely describes your relationship with the start-up?	(1) I am a founder of the start-up (2) I was hired by the start-up (3) I performed services for the start-up however I was not part of the core team (4) I have invested in this start-up (5) I have studied this start-up as part of a research project (6) I just happen to know this start-up

Table 3: Questions capturing precedents for technical debt

#	Category	Statements
1	Attitudes	The team does not see benefits from creating and maintaining good software architecture
2		The team does not see the benefits of following good coding practices
3		The team does not see benefits of throughout testing of the product/service
4	Pragmatism	Quick delivery of functionality is more important than good product design
5		Quick delivery of functionality is more important than the defect free product/service
6		Quick delivery of functionality is considered more important than good code
7	Communication	There is a communication gap between marketing/sales staff and developers regarding product/service features
8		There are communication issues within the team
9		Communication and collaboration within the development team regarding the product/service quality is insufficient
10		Communication and collaboration within the development team regarding the product/service architecture is insufficient
11	Skills	Communication and collaboration within the dev team is not sufficient to write clean code
12		There is a shortage of necessary competences in the team regarding product/service development
13		The team does not have required competences to create and maintain a good software architecture
14		The team does not have required competences to do good requirements engineering work
15	Resources	The team does not have the required competences to establish good testing practices
16		The team does not have the required competences to write good quality code
17		There are not enough resources and time to do proper requirements engineering work
18		There is a constant time pressure
19	Process	There is a constant shortage of resources
20		Requirements change often and cause difficulties in product/service development
21		Incremental changes to the product/service are unsystematic and degrades the architecture
22		Goals in the start-up are rather unclear and change often

Table 4: Questions capturing dimensions of technical debt

#	Category	Statements
1	Documentation debt	Even though the feature requirements are written down, they are difficult to understand and use in practice without consulting the source
2		Software architecture documentation is outdated, incomplete or non-existent
3	Architecture debt	Test cases are not documented in any systematic way
4		Initial product/service architecture has become outdated
5		Choice of product/service technologies is sub-optimal
6	Code debt	Looking for workarounds and patching the product/service architecture has become a routine task
7		The code contains multiple classes, methods or functions with similar functionality (code duplication)
8		The code contains unnecessary large and complex classes, methods or functions
9		The code contains classes without structure declaring long methods without parameters
10	Test debt	Many classes, methods and functions are not well commented
11		Test cases do not fully cover the product/service functionality
12		Manual testing is the primary practice to ensure that the product/service is defect free
13		Manual testing of the entire product/service is required to make sure that a release is defect free

Table 5: Questions capturing the extent of potential outcomes of technical debt

#	Category	Statements
1	Productivity	It is difficult to modify and add new features to the product/service
2		Modifying the product/service often takes more time than expected
3		Addressing problems in the existing code has become a routine chore
4		The code is difficult to modify or extend, productivity of the team is hindered
5		Making sure that the product/service is defect free has become a tedious routine task
6		Making sure that any existing features are not broken while developing a new feature takes significant amount of time
7	Quality	Product/service quality (performance, scalability, maintainability, security, robustness etc.) suffers due to issues in the software architecture
8		Product quality (Performance, scalability, maintainability, security, robustness etc.) suffers due to issues in the code
9		Changing or modifying the product/service features often takes more time than expected
10		Customers often report defects in the product/service that could have been captured before the release