

Observing best practices to address recurring concerns of product managers and product owners in large-scale agile development

Louis Leonardo Zschaler, 02.12.2019, Master Thesis Final Presentation

Chair of Software Engineering for Business Information Systems (sebis) Faculty of Informatics Technische Universität München wwwmatthes.in.tum.de

sebis

Motivation



- Research Approach
- Identification of Recurring Concerns
- 5
- Identification of Patterns and Anti-Patterns
- Exemplary Pattern



Motivation



51

Research Questions

- Research Approach
- Identification of Recurring Concerns
- 5
- Identification of Patterns and Anti-Patterns
- Exemplary Pattern



Motivation



Motivation





Agile development has been a proven concept in small companies for many years, bringing them various benefits. [3]



Large-Scale Many agile teams working on a project or product in a large scale sets a new form of complexity. [1,2]



Patterns Patterns are one way to ensure relevance of the research outcome, relating the findings to real concerns in the industry. [5]



Pattern Language Using the LSADPL as a structured set for providing patterns and filling this set with missing patterns. [7]

Introduction

A proven pattern template. [4]



Introduction



7

1





Research Approach

Motivation

- Identification of Recurring Concerns
- 5
- Identification of Patterns and Anti-Patterns
- Exemplary Pattern



Research Questions

Research Question 1

What are **recurring concerns** of Product Managers and Product Owners in Large-Scale Agile Development? **Research Question 2**

What are **best practices** for addressing recurring challenges of Product Managers and Product Owners in Large-Scale Agile Development?

Research Question 3

Which **anti-patterns** should Product Managers and Product Owners avoid in Large-Scale Agile Development?

Motivation



- Research Approach
- Identification of Recurring Concerns
- 5

3

- Identification of Patterns and Anti-Patterns
- Exemplary Pattern



Research Approach



Pattern-Based Research Design [5]



Research Approach – Structured Interviews

ID	Role	Interviewees Experience in large-scale agile development	Organization's Experience in large-scale agile development	Industry	Company Size
1	Product Owner	3 - 6 years	More than 6 years	Service Sector	More than 200.000
2	Product Owner	1 - 3 years	1 - 3 years	Telecommunications	5001 - 10.000
3	Product Manager	1 - 3 years	1 - 3 years	Service Sector	11 - 50
4	Product Owner	3 - 6 years	More than 6 years	Service Sector	More than 200.000
5	Product Manager	1 - 3 years	More than 6 years	Financial Services, Insurance, Retail	100.001 - 200.000
6	Product Owner	3 - 6 years	More than 6 years	Service Sector	251 - 500
7	Product Owner	1 - 3 years	1 - 3 years	Transport, Logistics	More than 200.000
8	Product Owner	1 - 3 years	1 - 3 years	IT, Technologie	1 - 10
9	Product Manager	1 - 3 years	3 - 6 years	Automotive	More than 200.000

Research Approach – Structured Interviews





Motivation



Research Questions

- Research Approach
- 4

Identification of Recurring Concerns



Identification of Patterns and Anti-Patterns





Identification of recurring Concerns





Identification of recurring Concerns

				· · +	c				
IVIa	anagemen	timpedii	ng the Ag	gile I rans	stormation	n; n = 9			
Ma	anaging pr	ovided s	olutions	of severa	il teams;	n = 1		_	
St	akeholders	s are not	enough	evolved	in the pro	jects pro	gress; n :	= 2	
Im	proving pr	ocesses	in agile o	organizat	ions; n =	3			
Er	notional in	npact of <i>i</i>	Agile Tra	nsformat	ions to er	nployees	s; n = 9		
Pa	itience dur	ing the A	Agile Trar	nsformati	on; n = 9				
Sp	read sgile	mindset	t through	entire or	ganisatio	n; n = 8			
Es	tablishing	a comm	ion under	standing	of agile t	hinking a	and practi	ces; n =	9
De	aling with	higher-le	evel man	agement	interfere	nces; n =	= 9		I
De	aling with	commur	nication c	aps with	stakehol	ders: n =	: 9		
Βι	ilding trus	t of stake	eholders	in agile r	ractices:	n = 9		1	
St	akeholders	s beina f	aced with	higher a	amount of	fmeeting	us: n = 9		
Co	ommunicat	ion char	nels fron	n higher	hierarchy	to lower	: n = 9		
Co	olission of	Program	Manage	ment an	d agile m	ethods: r	n = 9		
Cr	eating tea	m spirit a	and trust	among a	aile team	s: n = 9			
Fs	tablishing	a comm	on scope	for diffe	rent stake	eholder o	iroups: n	= 9	
De	aling with	internal	silos: n =				roupo, m	Ŭ	
De	aling with	increasi	na workle	had of ke	v stakehr	olders: n	= 9		
Te	Technical dependancies among teams: n = 8								
Ma	aintaining		ality amo	ng team	n = 6				_
Fe	timation	f comple	v doman	de/reque	s, n – 0 ete: n = 5				
E	tablish a c	ommon	vision of	the prod	1000000000000000000000000000000000000				
Dr	oduct Owr	or is the	vision or	une prou	the team $\frac{1}{2}$	n = 0			
E		managi	ng outon			0			
		office pr	litico: p -		ams, m –	9			
	Defining electroles and reaponsibilities n = 0								
	Defining clear roles and responsibilities; $n = 9$								
		nciease					Commun	ication, r	1-9
					proverne	m, n – 9			
		anageme		1, 11 – 9		- 0			
	Dealing with doubts in people about changes; $n = 9$								
All	griment of	self-org	anized te	ams; n =	0				
Sy	Stems Ini	iking; n	- /		La Tuana f	a man a fi a ra			
Ur	luerstandii	ng the de	emand to	r the Agi	ie Transfo	ormation	, n = 9		
0%	10%	20%	30%	40%	50%	60%	70%	80%	90%

Identified in Interviews

Identified in Literature

100%

Understanding the demand for the Agile Transformation; $n = 9$
Empowering agile teams to make decisions; $n = 9$
Providing sufficient tools and infrastructure for remote communications; n = 9
Dealing with incorrect practices of agile development; $n = 9$
Disconnect between corp. Strategy and execution; $n = 7$
Stuffing cross-functional teams equaly; n = 9
Synchronizing sprints in the large-scale agile development program; n = 9
Dealing with cultural differences between cross-shore agile teams; n = 9
Dealing with lacking sense of ownership responsibilities for developed services; n = 9
Dealing with closed mindedness; n = 9
Facilitating shared context and knowledge; $n = 9$
Coordinating geographically distributed agile teams; $n = 9$
Knowledge exchange between teams; $n = 2$
Product Owner lacking technical understanding; n = 2
Higher management dealing with a loss of control; n = 9
Stagnating continuous improvement process; n = 9
Comparability of Storypoints outside teams/projects; n = 9
Dealing with decreased predictability; $n = 9$
Considering required competencies when assigning teams to tasks; n = 9
Encouraging development teams to talk about tasks and impediments; n = 9
Dealing with loss of management control; n = 9
Dealing with black and white mindsets; n = 9
Dealing with lacking team cohesion at different locations; n = 9
Dealing with geographical distance between agile teams; n = 9
Creating a teamwork centric rewarding model; n = 9
Rearranging physical spaces; n = 9
Facilitating agile teams to participate at cross-shore meetings; n = 9
Synchronizing working hours of cross-shore agile teams ; n = 9
0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%

πп

Identification of Recurring Concerns

Concerns in categories

Culture & Mindset

- Communication & Coordination
- Enterprise Architecture
- Geographical Distribution
- Knowledge Management
- Methodology
- Project Management
- Quality Assurance
- Requirements Engineering
- Software Architecture
- Tooling



Concerns in hierarchal level



- Program Level
- Portfolio Level
- IT Organization Level
- Enterprise Level

Identification of recurring Concerns

Management impeding the Agile Transformation; n = 9	Understanding the demand for the Agile Transformation; n = 9
Managing provided solutions of several teams; n = 1	Empowering agile teams to make decisions; $n = 9$
Stakeholders are not enough evolved in the projects progress; n = 2	Providing sufficient tools and infrastructure for remote communications: $n = 9$
Improving processes in agile organizations; n = 3	Dealing with incorrect practices of agile development: $n = 9$
Emotional impact of Agile Transformations to employees; n = 9	Disconnect between correst Ctrategy and eventions n = 7
Patience during the Agile Transformation; n = 9	Disconnect between corp. Strategy and execution, II – 7
Spread sgile mindset through entire organisation; n = 8	Stuffing cross-functional teams equaly; $n = 9$
Establishing a common understanding of agile thinking and practices; n = 9	Synchronizing sprints in the large-scale agile development program; n = 9
Dealing with higher-level management interferences; n = 9	Dealing with cultural differences between cross-shore agile teams; n = 9
Dealing with communication gaps with stakeholders; n = 9	Dealing with lacking sense of ownership responsibilities for developed services; n = 9
Building trust of stakeholders in aglie practices; n = 9	Dealing with closed mindedness: $n = 9$
Stakenolders being faced with higher amount of meetings; n = 9	Eacilitating shared context and knowledge: $n = 9$
Collission of Dragram Management and agile methods: n = 0	Coordinating socgraphically distributed agile to real $n = 0$
Constitution of Program Management and agile feams: $n = 9$	Coordinating geographically distributed agrie teams, II – 9
Establishing a common scope for different stakeholder groups: $n = 9$	Knowledge exchange between teams; $n = 2$
Dealing with internal silos: $n = 9$	Product Owner lacking technical understanding; $n = 2$
Dealing with increasing workload of key stakeholders: $n = 9$	Higher management dealing with a loss of control; $n = 9$
Technical dependancies among teams: n = 8	Stagnating continuous improvement process; n = 9
Maintaining equal guality among teams: n = 6	Comparability of Storypoints outside teams/projects; $n = 9$
Estimation of complex demands/requests: n = 5	Dealing with decreased predictability: $n = 9$
Establish a common vision of the product; $n = 9$	Considering required competencies when assigning teams to tasks: $n = 9$
Product Owner is the only interface to the team; $n = 9$	Encouraging development teams to talk about tasks and impediments: $n = 0$
Forming and managing autonomous teams; n = 9	Encouraging development teams to taik about tasks and impediments, IT – 9
Dealing with office politics; n = 9	Dealing with loss of management control; $n = 9$
Defining clear roles and responsibilities; n = 9	Dealing with black and white mindsets; $n = 9$
Dealing with increased efforts by establishing inter-team communication; n = 9	Dealing with lacking team cohesion at different locations; $n = 9$
Establishing a culture of continuous improvement; $n = 9$	Dealing with geographical distance between agile teams; $n = 9$
Obtaining management buy-in; n = 9	Creating a teamwork centric rewarding model: n = 9
Dealing with doubts in people about changes; n = 9	Rearranging physical spaces: $n = 9$
Alignment of self-organized teams; n = 8	Eacilitating agile teams to participate at cross-shore meetings: $n = 0$
Systems Thinking; $n = 7$	Purchage agine teams to participate at 0.055-5101e Theetings, 11 – 9
Understanding the demand for the Agile Transformation; $h = 9$	Synchronizing working nours of cross-shore agile teams; n = 9
% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100	1% 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100

Motivation





- Research Approach
- Identification of Recurring Concerns



Identification of Patterns and Anti-Patterns



Exemplary Pattern



Identification of Patterns and Anti-Patterns





Identification of Patterns and Anti-Patterns

A-01 Don't let the Product Owner be the only interface to the stakeholder *	A-02 Don't think a change in too big steps ***	A-03 Don't instate a field specialist as Product Owner with no technical background *	A-04 Don't let teams work in the same constellation for too long **	CO-01 Structured coaching key stakeholders in entire organization **	CO-02 Cross-section architecture *	CO-03 Structured request for demand *
CO-04 Communication channel to maintain agile role within organization *	CO-05 Agile Governance *	CO-06 High-Level sprint planning *	CO-07 Dual-Track Agile *	M-01 Velocity Measurement *	M-02 Continuously changing the improvement method in retros *	M-03 Mapping storypoints to other KPI's *
M-04 Domain Driven Design **	M-05 Feature Teams ***	M-06 Change Backlog *	M-07 Flow analysis *	M-08 Nexus Sprint *	M-09 Strategical Backlog *	M-10 Weighted shortest job first *
M-11 Improvement Backlog *	M-12 Mob-Testing *	M-13 Magic estimate ***	M-14 T-shirt size estimation **	M-15 Agile Ninja *	P-01 Fully transparent agile project ***	P-02 The Agile Connector *
	P-03 OKR based incentives *	P-04 Culture of empowering decision making *	P-05 Intercultural team building *	P-06 Proactively involve key stakeholder in the progress with every increment *	V-03 Storymap **	
Anti-Pattern	Coordination-Patt	ern Methodology-	-Pattern Pi	rinciple Vi	ewpoint-Pattern	Pattern

Identification of Patterns and Anti-Patterns



٦Π

Motivation



- Research Approach
- Identification of Recurring Concerns
- 5

6

Identification of Patterns and Anti-Patterns

Exemplary Pattern



Conclusion

Exemplary Pattern – Magic Estimation



Pattern Overview				
Id	M-13			
Name	Magic Estimation			
Alias	_			
Summary	Magic estimate is an exercise for Scrum teams to do a rough estimation on a whole product backlog. It is very fast and therefor delivers very good results.			

Example

 The department for demand management at InsuranceAG noticed, that they spend a lot of time estimating new demands. A request for demand is raised whenever another department wants to start an IT supported project. But based on the estimated time and costs, many requests were withdrawn, which makes the work on a very detailed estimation useless.

Context

• Whenever there is a request for demand, it will not necessarily evolve into the start of a project.

Problem

- The following challenge is addressed by this M-Pattern:
 - C-116: Estimation of complex demands/requests

Exemplary Pattern – Magic Estimation

Forces

- Sometimes a request for demand is placed to gain more information about the complexity about the project without the intention of starting it.
- Performing full requirements engineering is the perfect basis for a very concrete estimation, but too much effort when it's not clear whether the projects comes to fruition.

Solution

- To perform the Magic Estimation a team of Requirements Engineers and experienced developers is needed.
- The Requirement Engineers must roughly break down the demand into epics or user stories, showing what most likely needs to be done to fulfil the demand. Based on this set, the experienced developers will now do a relative estimation, defining which epic requires more effort than another one, or less.
- They will start by defining a reference epic with a known effort and give it a fictive number (for instance story points). The estimation will be based on the Cohn scale [1,2,3,5,8,13,21,34,55,89,144], meaning every estimated item will be on this very scale and cannot be placed in between numbers.
- The developers will take turns placing a new item on the scale or move an existing one if they're not satisfied with its position. Items that are being moved too often will be taking out and be discussed at the end.



Exemplary Pattern – Magic Estimation

ПП

Variants

 Instead of the Fibonacci sequence, other fictive scale like the Cohn scale [0,1,2,3,5,8,13,20,40,100] can also be used. The procedure stays the same.

Consequences

- Benefits:
 - Easy to give a meaningful estimation in a short period of time
 - Deduce when all tasks in the backlog will be done
 - If some team members disagree, they can talk about the problems and solve them
 - Less effort in analysis
- Liabilities:
 - Less precise prediction

Motivation



- Research Approach
- Identification of Recurring Concerns
- 5
- Identification of Patterns and Anti-Patterns
- Exemplary Pattern



Conclusion

Conclusion



Key Findings

- 25 Concerns
- 29 Pattern-Candidates
- 4 Patterns

Future Work

- Findings on other stakeholders
- Validation of identified patterns
- Consolidation of all concerns, patterns, anti-patterns and pattern candidates

References

[1] Uludag, Ömer & Kleehaus, Martin & Caprano, Christoph & Matthes, Florian. (2018). Identifying and Structuring Challenges in Large-Scale Agile Development Based on a Structured Literature Review. 10.1109/EDOC.2018.00032.

[2] Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*, *119*, 87-108.

[3] Laanti, M., Salo, O., & Abrahamsson, P. (2011). Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. *Information and Software Technology*, 53(3), 276-290.

[4] Stepney, S. (2012, September). A pattern language for scientific simulations. In *Proceedings of the 2012 workshop on complex systems modelling and simulation, Orleans, France.*

 [5] Schneider, Alexander & Buckl, Sabine & Schweda, Christian & Matthes, Florian. (2013). Pattern-Based
Design Research – An Iterative Research Method Balancing Rigor and Relevance. 10.1007/978-3-642-38827-9_6.

[6] Henrik Kniberg (2016). Spotify Rhythm. Presentation at Agila Sverige.

[7] Omer Uluda g, Nina-Mareike Harders, and Florian Matthes. Documenting recurring concerns and patterns in large-scale agile development. 2019.

TLT sebis

B.Sc. Louis Leonardo Zschaler

Technische Universität München Faculty of Informatics Chair of Software Engineering for Business Information Systems

Boltzmannstraße 3 85748 Garching bei München

Tel +49.89.289. 17132 Fax +49.89.289.17136

matthes@in.tum.de wwwmatthes.in.tum.de



Backup







Allianz (1) DAIMLER







