Lean Software Development in Action

Andrea Janes • Giancarlo Succi

Lean Software Development in Action

Springer

Andrea Janes
Giancarlo Succi
Libera Università di Bolzano
Bolzano
Italy

ISBN 978-3-662-44178-7 ISBN 978-3-642-00503-9 (eBook)
DOI 10.1007/978-3-642-00503-9
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014953961

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

We wrote this book with the idea to show a practical implementation of Lean software development, gluing together well-proven tools to provide a way to develop Lean. The message this book wants to convey is the utilization of goal-oriented, automated measurement for the creation of a Lean organization and the facilitation of Lean software development.

Since its conception in the mid-1950s, Lean thinking has been very successful in manufacturing: it has helped organizations to focus on value-providing activities, to identify unnecessary ones, and therefore to increase the efficiency and effectiveness of the overall process. There are several proposals on how to translate Lean principles into software engineering practices, starting from the Agile Manifesto and the pioneering work of Kent Beck (see Chap. 4), of Mary and Tom Poppendieck (see Chap. 6), and of many others.

It is a fact that to be successful, a Lean orientation of software development has to go hand in hand with the business strategy of the company as a whole. To achieve such a goal, there are two interrelated aspects that require special attention: measurement and experience management.

Measurement means to describe real-world processes according to clearly defined rules to understand, control, and improve (see Chap. 9). Now, "Lean software company thinking" requires a comprehensive measurement program: measurement is a cornerstone of Lean thinking, as well evidenced by the founders of the Lean approach, such as Taiichi Ōno, James P. Womack, and Daniel T. Jones (see Chap. 1).

Managers of software companies have often perceived measurement as a waste of time, mostly because the measurement process required a lot of effort from developers and managers, and often the results of a measurement program could not be used to steer effectively the direction of the business. In the last decade, a new approach to software measurement has emerged, where, on one side, it has become evident that a successful software measurement program ought to be introduced in a company-wide measurement approach to promote process improvement, as discussed by the CMMI (see Chap. 3) and several other recent software process improvement initiatives, and, on the other side, the measurement process has become much less invasive and therefore effort consuming (see Chap. 9).

Measurement is a necessary step in systematic quality approaches like Six Sigma or in performance management instruments like the Balanced Scorecard (see Chap. 3). A methodical approach to measurement as proposed in this book will allow to introduce quality and performance management instruments that rely on it.

The second aspect requiring special attention is experience, i.e., valuable, stored, specific knowledge that was acquired in a previous problem-solving situation (see Chap. 8).

Software development organizations have always been concerned with collecting and institutionalizing the experience of its developers and engineers, so that it would not be too dependent on its key people and the turnover would not affect it too much. The experience of these years have shown that Lean software organizations are even more dependent on the skills of individual people.

It appears therefore critical to promote the institutionalization of experience across the entire company. The studies that the authors have done on the field have evidenced that the collected software measures can be extremely instrumental to achieve such a goal.

Altogether, in this book the authors provide the necessary knowledge to establish "Lean software company thinking" taking advantage of the most recent approaches to software measurement.

The main target group of the book consists of people working in the field of software engineering that want to understand how to obtain an efficient and effective software development process. This group includes developers, managers, and students following an M.Sc. curriculum in software engineering.

This book particularly applies to small and medium enterprises—the dominating organizational form in the European software sector—that do not have the resources to put in place the mentioned processes of collecting and institutionalizing the experience of its developers and engineers. This creates a high dependency on its key people, and turnover can severely challenge the ability to continue to provide a given product or service to the market. Additionally, not institutionalizing the own experiences means wasting an important opportunity for small and medium enterprises to maintain or improve their position in the market.

A comprehensive, company-wide measurement approach is exactly what (even very small) companies need to align the performed activities to the needs of the stakeholders, to the business strategy, etc. With the automatic, non-invasive measurement proposed in this book—this is what we mean by "Lean Software Development in action"—more companies will be able to optimize their software development process towards Lean.

The discussion on how to transfer Lean concepts into software development is still ongoing. To support this process, we aim to inspire others to follow our line of research to apply measurement to understand the effects of introduced changes in the software development process even at an early stage. Measurement helps to understand and to reason about the obtained desired and undesired changes. Moreover, it points out opportunities to improve.

We present our approach to Lean Software Development in three parts:

1. The first part illustrates what the term “Lean Production” means, why we think it is useful to transfer Lean concepts into software engineering, and which approaches exist to transfer the Lean concept into software engineering.
2. The second part illustrates the tools we use to achieve Lean Software Development: Non-invasive Measurement, the Goal Question Metric approach, and the Experience Factory.
3. The third part illustrates how we combined the different tools to enable Lean Thinking in software development.

Background Knowledge

Whenever we thought that some background knowledge is interesting or useful in a particular part of the text, we added it in a box like this.

A last but important note: throughout this book, wherever the masculine form is used, it applies to the feminine form as well.

Andrea Janes
Giancarlo Succi

Bozen, Italy
May 2014

Acronyms

ABC	Activity Based Costing
AJAX	Asynchronous JavaScript and XML
ANSI	American National Standards Institute
API	Application Programming Interface
PMI	Project Management Institute
AT&T	American Telephone and Telegraph Company
BC	Before Christ
CC	Cyclomatic Complexity
CD	Compact Disk
CIO	Chief Information Officer
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
DAD	Disciplined Agile Delivery
DMAIC	Define, Measure, Analyze, Improve, Control
DSDM	Dynamic Systems Development Method
ETL	Extract Transform Load
FSS	Flagship Software Services, a company in our fictional story
GQM	Goal Question Metrics
GUI	Graphical User Interface
HTML	HyperText Markup Language
IBM	International Business Machines
ICSE	International Conference on Software Engineering
ISO	International Organization for Standardization
IT	Information Technology
IQ	Intelligence Quotient
JSP	Java Server Pages
LOC	Lines Of Code
NASA	National Aeronautics and Space Administration
NPV	Net Present Value
PC	Process Costing
PDCA	Plan Do Check Act

PDSA	Plan Do Study Act
PSP	Personal Software Process
QIP	Quality in Practice
RCA	Resource Consumption Accounting
REST	REpresentational State Transfer
RFID	Radio-frequency identification
RAD	Rapid Application Development
ROI	Return on Investment
SOA	Service Oriented Architecture
SWOT	Strengths, Weaknesses, Opportunities, and Threats
UML	Unified Modeling Language
VP	Vice President
WMC	Weighted Methods per Class
WWI	World War I
WWII	World War II
XP	Extreme Programming
XXX	The boss in our fictional story

Contents