

UiO : Department of Informatics
University of Oslo

Do Work in Progress (WIP) - Limit in Agile Software Development Matter?

Truls Skeie
Master's Thesis Autumn 2013



Do Work in Progress (WIP) - Limit in Agile Software Development Matter?

Truls Skeie

18th December 2013

Abstract

Contents

I	Introduction	1
1	Background	3
1.1	Kanban	3
1.2	Scrum	4
II	The project	5
2	Planning the project	7
III	Conclusion	9
3	Results	11

List of Figures

List of Tables

Preface

Part I

Introduction

Chapter 1

Background

In the field of WIP and if WIP matters in software development, lacks proper research, but Giulio Concas and Hongyu Zhang has done research on the difference between limit WIP and unlimited WIP (Concas et al., 2013).

Work in progress is a tool that helps teams limits their tasks by specifying how many tasks a developer can be assigned to at once. WIP helps team to reduce overhead, decrease leadtime and increase throughput

How to find the best WIP in a given interval and context also lacks proper research, but in manufacture business some research has been done. Taho Yanga, Hsin-Pin Fub, Kuang-Yi Yanga stated that WIP could be defined as; $WIP = \text{cycle time} * \text{throughput rate}$ in manufacture business (Taho Yanga and Yanga, 2007).

1.1 Kanban

"We can define Kanban software process as a WIP limited pull system visualized by the Kanban board" (Anderson et al., 2011).

Kanban system focus on;

- Continuous flow of work
- No fixed iterations or sprints
- Work is delivered when it's done
- Teams only work on few tasks at the time specified by the limit WIP

- Make constant flow of released tasks

(Anderson et al., 2011).

Toyota production system was introduced Kanban during late 1940s and early 1950s in order to catch up with the American car industry (Ono, 1988). In the last ten years software Development Company have started to implement agile methods and Kanban is one of them (Conboy, 2009). Kanban splits one big problem into many small pieces of problems.

When the small pieces are in order, the problems are put up on the Kanban-board to visualize the problems and see potential bottlenecks. When people started to understand Kanban, they easily discovered where the bottlenecks were, and started to help where the bottlenecks were (Shinkle, 2009).

One of the most important people in Kanban software development, David Anderson also referred to as "father of Kanban in the software development industry" (Gupta, 2013), author of Kanban: Successful Evolutionary Change for Your Technology Business once stated "If you think that there was Capability Maturity Model Integration, there was Rational Unified Process, there was Extreme Programming and there was Scrum, Kanban is the next thing in that succession." (Leonardo Campos, 2013)

More and more software projects adapt to Kanban, and this is one of the reasons why this thesis will focus on Kanban and WIP. Kanban is one of the agile method in the wind these days, and is used with Lean Software development which is one of the fastest growing approaches in software development (Anderson et al., 2011)

One of the main difference between Scrum and Kanban is estimation, in simulation of software maintenance process, with and without a work-in-process limit (Concas et al., 2013) estimation was defined to be the main source of waste. In their research, they find out, if they let the developers work with small tasks at time and not be interrupted, they will be more effective. The developers in this case was interrupted when they were assigned to estimate tasks. The research groups decided to implement Lean-Kanban, which includes minimizing waste, which meant estimation for this case. After implementing Lean-Kanban the team's increased the ability to perform work, lower the lead time and meet the production dates.

1.2 Scrum

Scrum (Alliance, 2012)

Part II

The project

Chapter 2

Planning the project

Part III

Conclusion

Chapter 3

Results

Bibliography

- Alliance, Scrum (2012). 'Scrum, A description'. In:
- Anderson, David et al. (2011). 'Studying Lean-Kanban Approach Using Software Process Simulation'. In: *Agile Processes in Software Engineering and Extreme Programming*. Ed. by Alberto Sillitti et al. Vol. 77. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, pp. 12–26. ISBN: 978-3-642-20676-4. DOI: 10.1007/978-3-642-20677-1_2.
- Conboy, Kieran (2009). 'Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development'. In: *Information Systems Research* 20.3, pp. 329–354. DOI: 10.1287/isre.1090.0236.
- Concas, Giulio et al. (2013). 'Simulation of software maintenance process, with and without a work-in-process limit'. In: *Journal of Software: Evolution and Process* 25.12, pp. 1225–1248. ISSN: 2047-7481. DOI: 10.1002/smrv.1599.
- Gupta, Vikram (May 2013). *InfoQ Interviews David J. Anderson at Lean Kanban 2013 Conference*. URL: http://www.infoq.com/articles/David_Anderson_Lean_Kanban_2013_Conference_Interview (visited on 01/10/2013).
- Leonardo Campos Rafael Buzon, Eric Fer (Mar. 2013). *Kanban Pioneer: Interview with David J. Anderson*. URL: <http://www.infoq.com/articles/David-Anderson-Kanban/> (visited on 30/09/2013).
- Ono, Taiichi (1988). *Toyota production system: beyond large-scale production*. Productivity press.
- Shinkle, C.M. (2009). 'Applying the Dreyfus Model of Skill Acquisition to the Adoption of Kanban Systems at Software Engineering Professionals (SEP)'. In: *Agile Conference, 2009. AGILE '09*. Pp. 186–191. DOI: 10.1109/AGILE.2009.25.
- Taho Yanga, Hsin-Pin Fub and Kuang-Yi Yanga (2007). 'An evolutionary-simulation approach for the optimization of multi-constant work-in-process strategy'. In: 107, pp. 104–114.