

HOWTO: start Sprints without pain

Marina Arefyeva

HOWTO: start Sprints without pain

Marina Arefyeva

This book is for sale at <http://leanpub.com/sprintswithoutpain>

This version was published on 2015-05-04



Leanpub

This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

©2015 Marina Arefyeva

Contents

About Authour	1
BIO	2
Fear	2
Challanges	4
Agility	5
Kanban versus Sprints	8
QA versus support's Kanban	9
Developer's Kanban	9
Standups	10
Sprints or Kanban?	11
Getting started	12
Beeng agile	12
How to use this book	13
Define the current state of process	15
Understand your role on project	16
Development team lead or senior developer	16
Developer on project with small amount of developers	16
Project manager or product manager	16
Company's CEO, founder, etc.	17
QA Lead	17
Design Lead or other leads	17

CONTENTS

Exersices, level 1 18

1.1. First exercise - Project Structure 18

1.2. Add internal and external customers â€œorders flow
to your structure 19

1.3. Select your personal metrics of success 20

1.4. Plan structure for mail or meeting with internal
customers, but don't send it. 20

Work with objections 21

1. Talk to your boss 21

2. Talk to another people that can decide 21

3. Talk to internal and external customers 22

4. Tell anyone you see your idea 22

5. Find partner in process improvement 22

6. Define sub team that will be first 22

Make survey to get opinions 23

Exercises, level 2 24

2.1. Create your own survey draft 24

2.2. Correct and test the survey draft 26

About Authour

I'm now free Agile consultant and startup CEO from Russia with 10 years of professional experience on different parts of IT and still twenty six years old woman.

I often read that women get less success at work than men or get less salary on the same place. I saw one potential reason: fear.

It may be strange concurrence, but I went from fear to do something wrong to challenges only after year on Karate training. May be that's reason, because men in Russia go to army for a year, but woman usually doesn't have such experience and can be afraid to do something and afraid to fail.

When you know that feeling of fight, especially on fighting sports, you are not afraid of it and you wait for this event and prepare to it to win first place.

I think this is some kind of agile spirit - to be ready for virtual fight every day. Each problem is your competitor, but you already know that one day will be fight and you are prepared and know if you win or lose this fight, there will be another fight and cases from this will help you prepare to move forward even if you've lost something that time.

I divided my path to three parts: fear, challenges and agility.

BIO

Fear

Fear of misunderstanding

When I was sixteen and studied at College of Information technologies of Novosibirsk State University I decided that old enough to live with friends on flat but not with parents. To prove this was necessary to find job that would give enough money to pay for food and other necessities, father paid my rent part and i did not want to ask more. My first job was in pizza bar for half of year and first IT job was in 2005 as a QA engineer.

I remember how it was difficult to fight complexes inside me. I thought they all are smarter just because older. I was afraid to be useless and to lose job for first four months. But even that days there were people that don't understand what are they doing and I had a lot of work to report problems even on technical design level.

It was just a place with cheap workers, and founders were happy to have some people that work for \$300 together with good qualified engineers. I decided to find something more after graduating from college and did it.

Many of people from that team still stay at the same stage for years, with almost the same salary on the same job.

If you don't improve yourself you cannot improve your work skills and job position. Changes bring fear, and fear can win.

Fear of failure

Once we with friend had chance to start own outsource project with really promising customer, we did it. But I still was afraid of many things.

No one wanted to write Linux kernel modules. They thought it's impossibly difficult job and did not even try. I had to hire my friends from college that did not know word "fear" at all.

All the difficulty there was to start. Now i cannot understand what was so scary in writing of first lines of code on project but noone wanted. So I did it.

I remember words *insmod* and *rmmod* that started and finished our module and it was as simple as my first hello world program.

We made project first demo before deadline, so our client found investors. Now I realize that it is really fantastic. Did not see any project after that that was made up to demo at necessary time.

After that I went out from that business and focused on study combined with half day work.

My next job was exact opposite example: I wrote automated tests for one installer for half a year and still work was not finished - manager wanted 100% cover of installing cases.

Fear of changes

In 2008 I started work at company that does not exist now: Linux XP. Readers might remember Windows XP. Idea was to take Fedora Linux and make it look just like Window XP. Why? Problem of computer viruses disappear and no need to teach low qualified computer users on Russian government teams to use new OS. But company did not win competition for government Linux in 2010 and had to disappear. Team created new company and continued to work on another sector.

I wanted to leave this position year earlier but was afraid because of economics crisis and used this chance to find new work with new challenges.

Challenges

Process creation Challenge

I was first QA at that team named Playtox. The team was promising and had already working service that gave a lot of money before. *They did not know how to DO Agile but WAS agile.* When I started team's planning meetings instead of missing CEO and at the end founder opened the door and said "Hi All! We are doing all wrong! Our users fail to pass from 5th to 6th level in game!!" and we started iteration planning again. This is one of the fastest team I worked in.

Once I returned from Agile the training and told designers about another way to make fast prototypes: just make it on paper, they were happy and then for 2 month their tables became full of papers. After that I took some part of CEO's duties cause CEO went to another company, and led technical department for a month and then explained new CEO how all IT process work. It was funny even though we had specific founder's communication processes. I think they just wanted all to work even faster and was frustrated, but too often angry.

On that work I first saw as developer made Kanban process to his team and everyone was OK with it and process went even faster, when each feature was in separated brunch and had own ticket on Kanban board and own release date that could change if another feature went to fast track.

Team fast growing challenge

One day Iâ€™ve got call with question “how about working at Yandex?”. Yandex (NASDAQ YNDX) is biggest search engine in Russia and 4-Th biggest in the world. They have one big challenge everyday - their market share on local market is over 60% and Google is on second position. Of course, Google don’t like it.

First of all this team was in St.Petersburg, that is more than 4 000 kilometers away from hometown. Next is that I was 23 years old new engineering lead in team of professionals and was able to prove that I can make better process of quality assurance group then they had before. Growing team from 7 engineers to 30 at one year was good process building experience.

I went to team of Yandex.bar product that gave Yandex services to browser, but when I was first day at work my boss told me that my job will be super-secret project - build QA team for Yandex.Browser from almost zero. This product was not released at that days and we worked hard to make first release at October, 1, 2012.

After more than year at this position I understood that want to change development and support process.

Agility

I changed job position to project manager and duties to process improvement in QA, support and St.Petersburg part of development team. And since that time I started to practice all the process improving information that was able to find. Because my challenge was to make more than twenty developers agile. This was the only way to prove to myself that I can change a process and manage middle sized teams, more than forty persons. I was still unsure but already was not afraid.

Goal of this period was to experiment with St.Petersburg part of

desktop Browser development team more effective in communications with management office in Moscow.

I had no goal to make them write code better or faster. Agile does not solve such problems directly. I did not have my own product team, o now was manager only for myself and abstract part of project.

People in company asked “Why you left QA management after so much years in it?” was “When you manage people they ask a lot of questions every day that you have to manage. When you manage project documentation as analyst wiki pages are silent and it’s boring. But when you manage process you run from one person to another and ask them to do something, and if you won’t they are even more happy. So sometimes you can take time to think and just watch how process go. It’s perfect for me.”

On this position I understood management rule: after changes made people need some time to adopt and person who made changes need to measure success of this changes.

For manager that is responsible for process changes not to make too many changes people don’t understand and don’t approve should be several experiments in separated groups of people. In this book I will tell about QA, support, developers and even designers as different target groups. But they of course work together and have single process using different ways of thinking.

We we overloaded by 300+ tasks on each developer’s group and every task owner wanted it to be done as soon as possible. Sprint planning solves one problem: all internal customers or external customer know exactly what tasks can be made next two weeks, and you need to make some preparation for work as if just select ticket and starting doing it.

Sprints seems to be more difficult in implementation than Kanban. But it does not, because you will need more self confident people on both sides, that understand process and understand what they should do on Kanban. If you have five engineers and twenty tickets

and able to control incoming flow it's almost OK.

You should understand all problems you will solve and either all problems and additional work you will create using it.

First of all I wanted to make Kanban process to have soft changes in team. And that was a kind of mistake, not Kanban as practice, but choosing this instrument on that stage processes.

If I only would know that implementation of sprints can be such easy process as it was on third and all other teams we repeated process, I would finish Agile challenge not in one and half year for more than ten teams, but in half of year.

When process grew up and became self changing my work on this task was finished went to Agile Days Russia 2015, where told on beginners section how to start sprints.

After positive feedback from that conference which was quiet surprising, I decided to write this book.

Please do not hesitate to contact me on email: marina.arefieva@gmail.com or twitter: @omnisa if you have any questions about sprints or process management.

Kanban versus Sprints

##Support versus Kanban Most simple step was creating second line of support engineers. This engineer was happy to get Kanban. Other engineers did not think about it and used support tools as usual with sending some tickets to JIRA and getting mail notifications about status changes. How it works:

1. Support engineer of first line get request that he or she doesn't know how to manage.
2. Requests go to JIRA tracker to second line engineer as a potential bug.
3. If bug is duplicating already existing but, engineer increase duplicates count for original bug and closes this ticket.
4. If problem is new he can put it to custom status, e.g. "need more info from user".
5. After reply of first engineer second line support engineer should decide: if ticket can be reproduced it goes to management, or goes to QA engineers to reproduce.

Four duplicates made ticket very valuable and ten made it candidate to blockers(max priority in JIRA).

This process allows understanding of the top problems that have users on beta stage during two days after launch and works without any additional planning.

All went well for eight months until any questions from another manager in project management office showed that we forgot to tell about it officially. We discussed it again with big team but only changes that made was amount of duplicates that make ticket look like reason to make hot fix, it became twenty instead of ten.

QA versus support's Kanban

Another side of feedback process from users is QA group that should check reports that are not so easily reproduced but still can be valuable.

Before we made Kanban board all tickets had OPEN state but really was in different states. Several was just open, another waited for users reply, and only third group was ready for test and had information translated from user's language to technical description.

We made status "READY FOR TEST" and QA engineers started to look this type of tickets first. They also had their specific status for forwarding question to support "Need info", and it was another status "need info" than supports inside.

In this case we've got really working Kanban system with Green Hopper Kanban board and statuses that was easy to determine.

Developer's Kanban

At the same time I was responsible for increasing productivity of communications between developers in St. Petersburg and managers in Moscow. We had everything we need to communicate between offices: phone on each table, meeting rooms with video connection, and travels to Moscow anytime we need. But process still was unmeasurable and somehow random.

We started from Kanban board and put it to TV on our room. First thing that changed – we became advanced users of TV on Linux. We won together and got access to internal network where were our Kanban board. Nothing else changed from first two weeks. Why? Because every change in process we wanted to make had questions from developers' bosses one or two levels up. One of them really was in opposite side and wanted all to stay as it was before. May be I did not understand how to deal with it. But they did not decline

all changes. It just went slow and through a lot of e-mails and conversations. After several months of such standing on one place experiment I've got new duty: group of two developers that wanted me to represent them for other project managers. They made their own sprints for two persons and two weeks and when I came it was already 19th sprint. They used the same JIRA plugin Green Hopper but Agile Board instead of Kanban board. Later I occasionally found Kanban board of their previous project manager and understood how did he see situation spending a little time.

Standups

Standups in usual Russian IT companies is thing that I hate more than anything at work. You stand for twenty or even more minutes and hear how people try to remember what they did yesterday. Almost no one thinks that it's time for co-working and think that it's part of total control about each their step that was made. Some people tell that they will go to another job if someone will make standup in their team.

Useful standups I saw in Kanban team that focused on what went wrong yesterday and each person told a topic to discuss with others.

I had challenge to Once our Kanban team went to standups of another team that practiced sprints. They found it useless and I was QA also loves this standups because can talk to all developers of her group what went wrong and need to be fixed in several words, not tracker. But there were no design department on thees meetings, cause we did not have separated designer for group and this questions was always difficult to solve.

Sprints or Kanban?

If You want to start sprints on short time and avoid most of typical mistakes I made on this way, follow instructions below on this book.

If You have more self-motivated people and don't need to rush as a startup or if want sprints just because someone told that he need you to have sprints, just remember what's the difference between *doing agile* and *being agile* and find your own way.

If You decided to start, please carefully make exercises at the end of chapter. Practice is better helper than words of book.

Please do not hesitate to contact me on email: marina.arefieva@gmail.com or twitter: @omnisa if you have any questions about sprints or process management.

Getting started

I hope you started reading this book with serious process changing goal, because I will ask you to make some exercises to be sure you are going to your goals. Please contact me anytime with questions or exercises answers. I need feedback to improve this information too.

But don't be afraid if your team become less productive for some time. It's cost of innovation and if all goes good it will increase after two or three months.

Beeng agile

What's the difference between **doing agile** and **BEEN agile**? Do things that *You* need to do with the way that matches better to your goals. SCRUM principles are very difficult for beginners to remember.

Let's imagine that You are driving the car with friend and he or she screams "Be careful! There's a car! Didn't you see?!". Sometimes you say *thank you* because it saved your car and your health and sometimes you hate this person.

Being agile means going your own way, not just repeating what made others. Also it means to be ready for changes any time, like you are ready to react for anything on road when driving.

Please contact me in [twitter](#)¹ if you want to help me in this project.

¹<https://twitter.com/omnisa>

How to use this book

This book will contain tests and surveys for understanding that readers are doing well.

Please read only one chapter at one time and make exercises before reading another.

Find answers together with your team on sprints retrospectives for better results.

For starting sprints from zero

All adaptation process for sprints usually lasts for 4-6 months, even if You already use Kanban.

Each chapter matches steps of process improvement and if You did not start yet, just go from first to last.

For teams with chronic problems to fix

You might be interested in correcting of team behavior in several things. Select chapters from list and begin from exercises. work on one chapter at one time.

For teams that not sure that they need sprints

You should decide it first. Because any changes will not work if you are not sure.

Also try to answer several questions:

1. Do you have several managers on single team?
2. Do you want often releases?
3. Do you have tasks to do more than people in team can?

4. Does your client know what development made last 2 weeks?
5. Is your process clear for other departments or client?

If you want to improve something one of possible choices is *sprints*, but don't try to break current process, you can implement it softly and I'll try to tell you how.

Define the current state of process

Before you start making changes look around and carefully write down current state of process.

Here we will talk about teams on software or hardware development, but Iâ€™m sure that it can be adapted to any other team if this team will be ready to changes.

Understand your role on project

Development team lead or senior developer

If you have this role you have almost all chances to start successfully. But talk to management first to agree to give you majority of tasks once a period and only extra tasks during sprint. Talk to managers first.

If you are not, go to talk to developer's lead before.

Developer on project with small amount of developers

Try to take this to you but find partner inside of managers first. Give him this book to speak on the same language. And try. It may be difficult but definitely interesting. If will not. Just find someone who will be instead of you.

Project manager or product manager

Depends on how strong this role's position, you need to talk to senior management, developer's lead, qa lead, designer's or any role's lead if this leads can block your decision.

Not doing this you will also can improve process, but it may take much more time that you want. Up to two years for projects with several sub teams.

Company's CEO, founder, etc.

Just find person that will represent you on current team to process changes with people. Some people will really be afraid of giving feedback.

Better role is developer's managers or project managers, but should be agile activist. You can send several potential activists to training and look who will return more engaged.

QA Lead

I was one of that QA leads that decided that good processes is also part of quality. But to understand all processes i went to project manager's role on the same project that is responsible for agile process organization. It was difficult because developer's leads still remembered me as QA. But it was very nice to hear after that from them that my idea really works.

Simple way is to inspire your project manager or managers to make such changes.

Design Lead or other leads

You should act according to the situation. Main goal is to be sure that main project activities are also agile, but you can make your own sprints that will include only designers, for example, it's also good decision. Otherwise find partner from project or product managers.

Exersices, level 1

1.1. First exercise - Project Structure

You should draw how your team structure look like.

Count developers, QA's, analytics, designers, managers, any other roles.

Write down the lines of connections between team members.

Think what changes you will make to project structure for people to be ready to be

Don't look next page until you finish this exercise.

Check: Maximum win is if you found already existing structure information and corrected it to actual state.

Good if you draw it and asked others to review before thinking that it's ready.

If your structure full and measure how many people you did not even know about. If less than 20% it's ok. Nobody can know anything, that's why we work in groups.

Not so good if you sure you know all yourself and did not check information. Check it now.

Bad only if you did not find even 20% of team members to write down them to list. For example, many people think that QA is another opposite team and don't add them to their vision.

1.2. Add internal and external customers ~ orders flow to your structure

Find people who are internal customers to team and generate tasks and ideas inside. This list will be useful for announce about starting sprints.

Find out what teams make work for more than one internal customer. Usually it is company's helpdesk or team that supports infrastructure in IT companies.

Suggest to your boss to discuss structure to make sure you know all customers. If your team works directly with external customers list people that work with them as internal providers of customers.

Check: you should find at least one such team part that has several customers or providers. This team may be listed as non-effective by managers if they don't contact to each other.

1.3. Select your personal metrics of success

When you start some process changes you are an enemy for any conservator. You should know that you are doing right things and talk about it with conservators with proving details. What do you thing will be good result for process changes?

Check: Solving some problems and not making more problems at that time - it's an abstract answer. Your answer goes from reasons you started to read this book. I hope it is not because your boss said "I need sprints because my boss said me that your team needs sprints".

Each problem should have measurable quantity and goal. This will be your metrics.

1.4. Plan structure for mail or meeting with internal customers, but don't send it.

One bad cause why sprints are difficult to start is a lot of organization mistakes at very beginning that causes lack of information and people just don't understand what you need from them and how to interact with team after changes.

Check: if you read this it's already OK. You finished reading chapter and hope made all exercises. LEVEL 2 - OBJECTIONS.

Work with objections

1. Talk to your boss

If you already have team's structure, personal metrics of team success, list of customers, and draft of letter to internal customer, lets start next level - work with objections. *If not, please made exersices from previous chapter.*

Tell your boss what problems in processes exactly you see and how you suggest to fix it.

It's main point before you *start expansion*. Otherwise you could spend a lot of time fighting against them and process improvement may become a cold war.

2. Talk to another people that can decide

When you are sure that your boss is ok with it, talk to leading roles on your project together. They should know about the idea from first hands and be able to make changes in your process vision.

There are so many things to coordinate that one person cannot manage it even in team from 10 persons.

And, you will want to go to vacation once, and if your leads are not your changes partners day when you go to vacation can become last day of sprints.

3. Talk to internal and external customers

They should understand that there will be changes and should be ready not to block your activities. One most important phrase they should be ready to hear is *“we will discuss it on next sprint planning meeting”*. Next phrase is *“If You did not prepare task to sprint planning enough it cannot go to sprint.”*

4. Tell anyone you see your idea

It sounds like startup inside of existing process. You are going to implement new way of thinking. Start from yourself and become more open to talk what you want to change. Talk to other people in company even if they are not in your department. Talk even to your friends if they are ok to talk about work.

5. Find partner in process improvement

You definitely need partner in this big process, you will solve a lot of problems and may sometimes feel that whole team is against you cause people are usually against changes. And you will need vacation once.

6. Define sub team that will be first

Select sub team to start. Even if you have team less than 10 persons you can divide it to two separated parts to measure changes in process like A/B testing.

Understand what time is better to start active changes according to their opinion and opinion of your boss.

Make survey to get opinions

Review anyone's opinion in selected group about current processes, and review team's engagement level.

Most simple way to measure opinions difference is making survey. For better results please make survey using *Exercises* of this chapter before running survey to all members of team.

OK, now we are ready to start our way to successful sprints. Let's do this important thing together using exercises 2.1. - 2.4.

Exercises, level 2

2.1. Create your own survey draft

But don't start. Just prepare.

Write down there that answers will be not in secret to others including name of respondent. People should understand what they write is what they share to each other.

Use your own list of questions, but look to mine before:

1. Q: How do You like our process? A:: select 1 to 10, where 10 is maximum of possible happiness.
2. Q: What is the best practice we have? A: paragraph edit
3. Q: What is the worst thing in process? A: select from prepared list of problems. For example build system problems, lack of testing, not enough time for free refactoring or something like that you know people speak around.
4. Q: Did you have bad experience about agile, scrum or sprints in past? A: paragraph text
5. Q: What do you think about daily standup meetings? A: select from several cases: "I love it and know how to use results after", "I like it", "I'm ok with it", "I hate it".
6. Q: what do you think about planning poker? A: select from: "5 - I love it because of better prediction", "4 - I like it because it's funny", "3 - I'm ok with it", "2 - I don't know what is it", "1 - I don't care", "0 - I hate it cause it's useless". Here you can send to results from digits and calculate median value.

7. Q: Do you want to participate agile processes workgroup? A: possible choices: "Sure", "Yes", "May be", "Don't want but can", "No".
8. Q: Do you want to suggest something to agile workgroup? A: paragraph text
9. Q: Do you remember any of agile principles from it's manifesto? A: paragraph text.
10. Q: Do you want additional training to work in agile team? A: "Definitely, yes. You all are from Mars.", "Not sure", "No thank you", "I can make training myself for our team".

Don't look next page until you finish this exercise.

Did you make survey exactly the same as is on page before?

YES

If yes, you failed. This is only example. Remember: “been agile, not doing”. This survey questions was created in 10 minutes just to be looking good for first view. But what will give results to you?

NO

Good start. Even if you changed only one field it is already result of your decisions.

2.2. Correct and test the survey draft

When I made my first survey I thought it's not serious, and i don't need to go deeper. No, any person you will ask will spend at least 5 minutes to your questions. So don't waste their time with useless questions you don't know how to analyze.

Change all questions so that you can get info about average opinion not reading and translating to digits every answer. But one field for comments should definitely be. Or two: one for positive, another for negative.

Test your survey on your boss and correct it with him or her. Then test in on leads of each substructure you painted on *Exercise 1.1.*

If you had problems with creating survey, use any book about surveys.

Check: at least two persons answered seriously and you checked spelling mistakes.

OK, next is LEVEL 3 - SURVEY DAYS.

Please do not hesitate to contact me on email: marina.arefieva@gmail.com or twitter: @omnisa if you have any questions about sprints or process management.