Planning as a Social Event

Scaling Agile @ LEGO

Henrik Kniberg & Eik Thyrsted Brandsgård

December 2016

- What? A 150-person meeting for ~~2 days~~ 1 day?
- Yeah! Every second month. Works great.
- But why? And how?

Background

LEGO Digital Solutions (DS) is a group of 20 or so teams responsible for handling communication with kids and parents via whatever device they are using - computers, tablets, apps, wearables, VR, etc. We also look into future product development, how to embrace new technology, how to take the classical way of playing with toys and combining with something cool, like augmented reality, or ways of "scanning" a physical model and getting it into a game. Most teams are in Billund, Denmark, but we also have a bunch of teams in India.

LEGO, at it’s core, is not a digital company. It is an 80-year old 17000-person organization optimized for design and manufacturing and marketing of physical toys. Like most companies, LEGO is trying to adapt to the faster-paced digital world, a world where change is constant and agile development is becoming the norm.

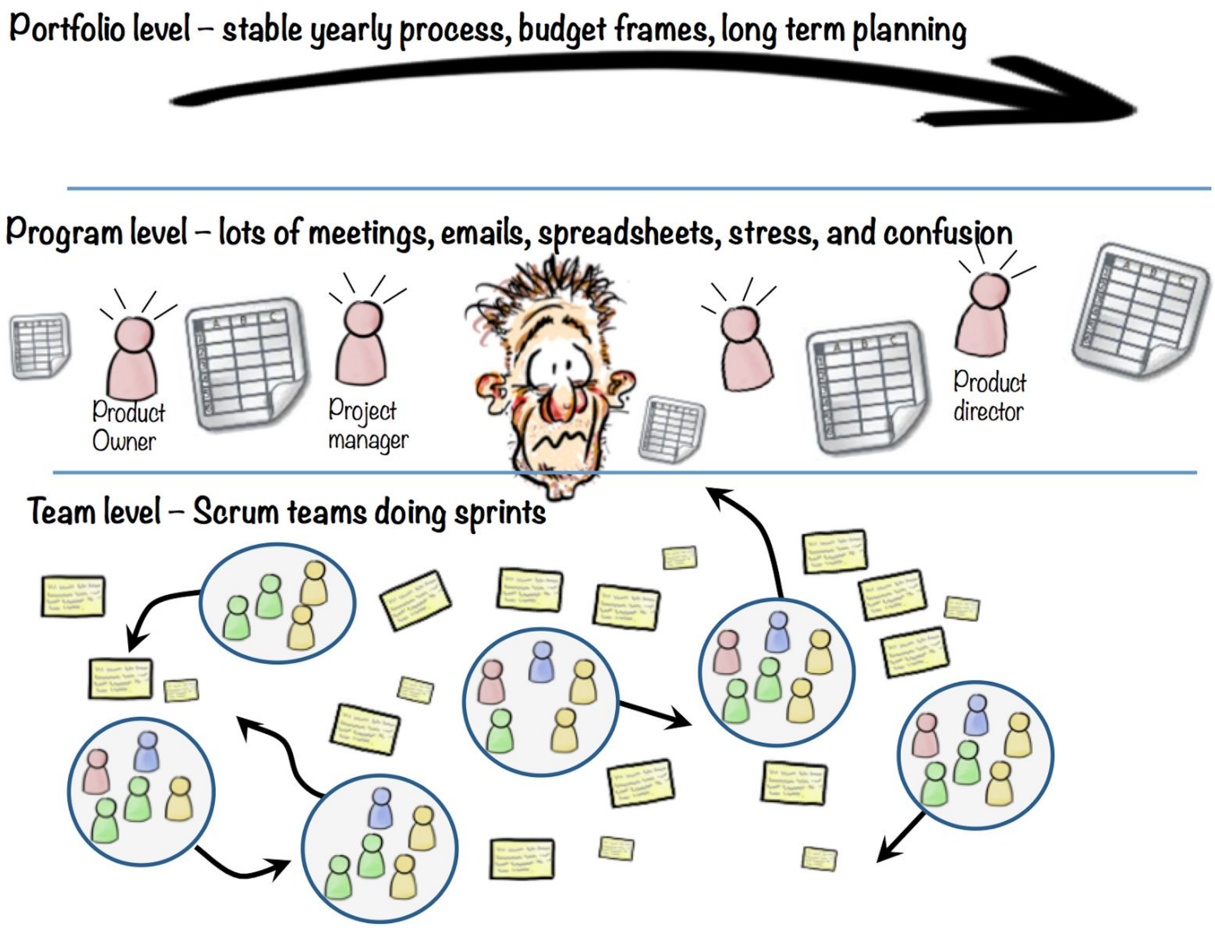
Digital Solutions is at the forefront of this change at LEGO, and we’re on a very interesting journey that we’d like to share with you!

The Problem

Back when we only had 5 or so development teams, planning and synchronization was pretty manageable. Teams and Product Owners could basically just talk to each other and figure out what to do. However as we grew to 15-20 teams things started getting more painful.

LEGO as a whole has a pretty good portfolio and budgeting process. Investment frames are negotiated on a yearly basis (X hours for project Y), while the actual content decisions are decoupled from the finance decisions, so departments can be pretty flexible about how to spend their efforts.

So in 2014 we had this stable portfolio management process at the top, and at the team level we had 15-20 teams doing Scrum and sprints. The problem was in the middle - the program level!


Portfolio level – stable yearly process, budget frames, long term planning
Program level – lots of meetings, emails, spreadsheets, stress, and confusion
Product Owner
Project manager
Product director
Team level – Scrum teams doing sprints

We were trying to use Scrum and work in an agile way, but on top of that we were basically a matrix organization doing projects. As product owner you spent almost all your time in meetings coordinating teams in order to get anything done, and eventually you wouldn't get what you needed anyway because other teams had other priorities.

As platform product owner you could have 10 people coming and asking for stuff, and you couldn't possibly deliver everything. So how do you decide which things are more important? Sometimes teams would end up building identical things - like, hey, wouldn't it be cool to also have a roller coaster in addition to the five merry-go-rounds we already have? Can't have too much of a good thing, or?

In summary, we were struggling with:

- **Cross-team alignment -** how to get teams moving in the same direction instead of stumbling over each other
- **Client collaboration** - how to set realistic expectations and satisfy the clients without over-committing
- **Release planning** - how to plan and prioritize work across multiple sprints, multiple teams, and multiple products

- **Platform development -** how to make sure we invest for the future and don't just implement a bunch of one-off solutions

As we neared the end of 2014, we decided to try something different.

The change

In January 2015 we got bold and started shaking things up. We morphed the entire DS organization into a team-of-teams, introduced a shared sprint cadence, decentralized syncronization and dependency management, and big-room planning events every 8 weeks. This has had a whole lot of positive effects, not only for DS, but also for other departments that we collaborate with.

So let's take a look at where we ended up after a year and a half of experimentation.

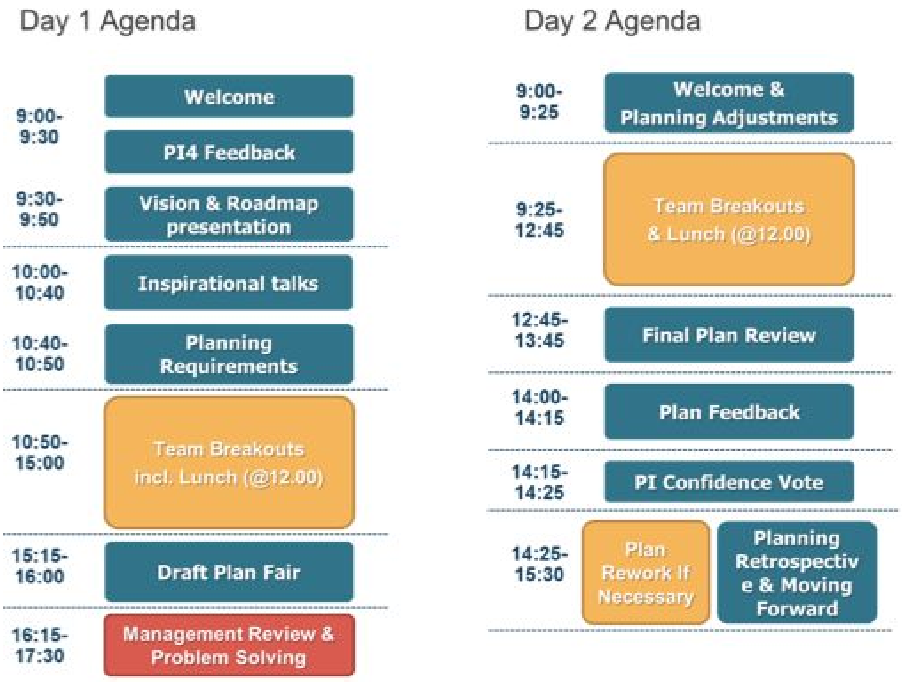
Disclaimer*: This is a journey in progress, not a journey completed. We keep evolving the model - for example in Q3 2016 we reduced the length of the big-room planning event to 1 day instead of 2 days. It's tempting to keep updating the article, but then it'll never get done, so let's just say this article is a snapshot of how things looked in Q2 2016 :o)*

A tour of Big Room Planning

It's Wednesday morning and you have been invited to this mysterious event, the "PI Planning" that people keep talking about. You know it stands or "Product Increment Planning", and that it is some kind of planning event that happens every 8 weeks. You step into a big gymnasium and see this:

Just kidding…. you see this:

Looking around, you see that all DS development teams are there, all managers, many of the clients and stakeholders, and a handful of curious visitors like yourself. A high level agenda is on the table.



Day 1 Agenda

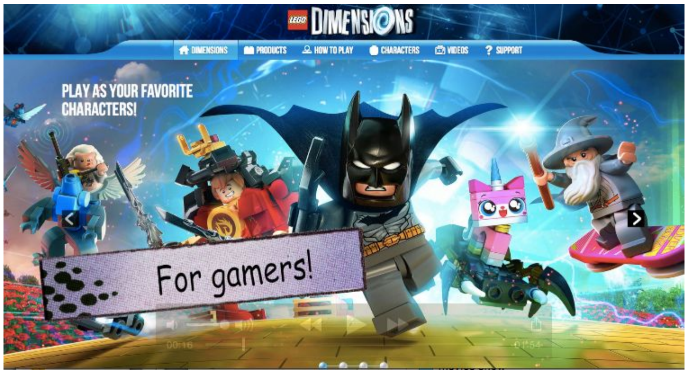
9:00-9:30	Welcome PI4 Feedback
9:30-9:50	Vision & Roadmap presentation
10:00-10:40	Inspirational talks
10:40-10:50	Planning Requirements
10:50-15:00	Team Breakouts incl. Lunch (@12.00)
15:15-16:00	Draft Plan Fair
16:15-17:30	Management Review & Problem Solving

Day 2 Agenda

9:00-9:25	Welcome & Planning Adjustments	
9:25-12:45	Team Breakouts & Lunch (@12.00)	
12:45-13:45	Final Plan Review	
14:00-14:15	Plan Feedback	
14:15-14:25	PI Confidence Vote	
14:25-15:30	Plan Rework If Necessary	Planning Retrospective & Moving Forward

Demo time!

After a few minutes, music starts blaring out and a fast-paced 5 minute video shows off highlights of stuff that has been released during the past 2 months, followed by cheerful applause.

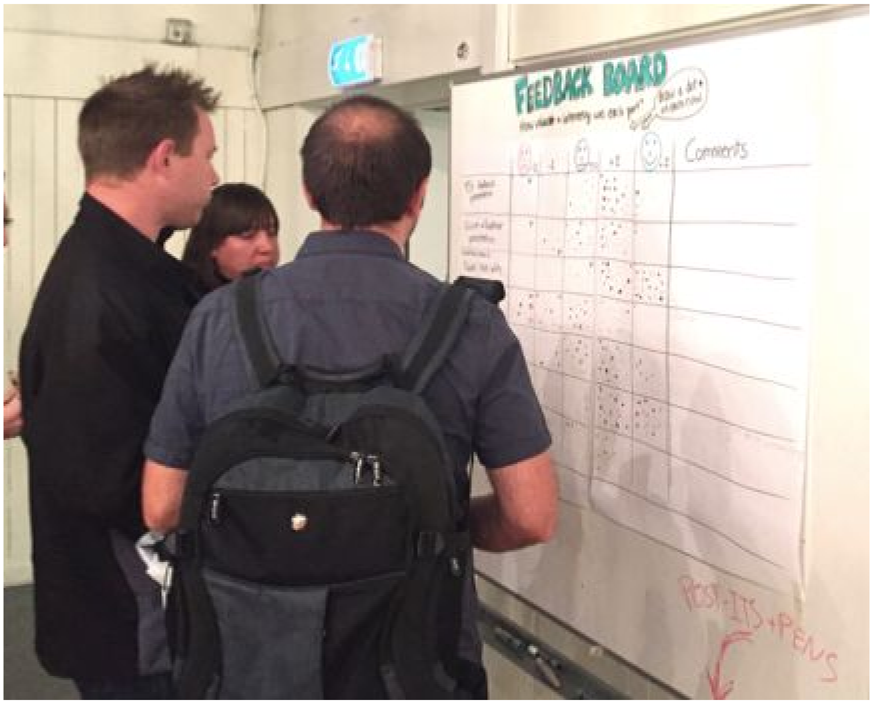


This doesn't replace the day-to-day feedback that teams get through sprint reviews and user testing and such. But it gives a nice overview, and a dose of inspiration. It reminds everyone about the purpose of what we do.

Lightning talks

Next a manager gets up for a couple of minutes, shares some thoughts about what's going on and what's important going forward. Then others get up and do a series of short inspirational talks on topics such as digital child safety, an upcoming hackathon, platform strategy, and other Hot Topics for the day. Someone demos how much easier it is to release stuff using the new platform, another runs a fun but silly Kahoot quiz.

By mid-morning the plenary stuff is done, and as people wander off for coffee they rate each session on a feedback board. How valuable and interesting was each talk? Scale 1-5.



- Joe: “Look, mostly 4s and 5s! Much better than last time.”
- Lisa: “Except that fourth topic though. Ugh!”
- Joe: “Yeah, that was probably only relevant to like ten people. I fell asleep.”

You see small comments to help improve the talks next time. A couple of people debate about whether we should have lightning talks at all or if it’s better to just launch into planning directly next time.

The facilitator turns to you. “Plenary sessions are like a time bomb, gotta keep things short and to-the-point or people tune out. When done right it can be a real energy boost though. Gives people a sense of context, which helps in the planning.”

Team breakouts

Next up: team breakouts. The room is alight with buzz as each teams work in parallel, figuring out the key things to deliver during the next couple of months. A high level plan, not too detailed, since plenty of unexpected stuff can happen in 8 weeks.

Each team has their own rollable whiteboard and planning board. Most team members are standing in front of the board discussing with each other, while some are roaming around to sync with other teams and stakeholders. Some are standing around the coffee and snack table discussing other random topics.

You notice that some teams and roles have team t-shirts or special hats, to make it easier to spot each other in the crowd. The Indian teams have representatives at the meeting, communicating back and forth between the people in the room and the rest of the team members in India.

Looks chaotic at first glance, but the energy level is pretty high. Although some seem a bit stressed or bored, the majority look like they are having a good time and being effective. Plenty of spontaneous conversations and laughter, and new connections being formed, existing connections strengthened. It's clear that they have done this before and are comfortable with it.

This part of the event takes up most of the day, and it feels very much like an Open Space event (if you've been to one of those). Just like in an Open Space event, the single most imporant rule is Law of 2 feet.

> ***Law of 2 feet:*** *If you aren't learning or contributing or having fun where you stand right now, use your 2 feet and go somewhere where you can learn or contribute or have fun. (OK we added the "have fun" part, it's not officially part of Open Space.)*

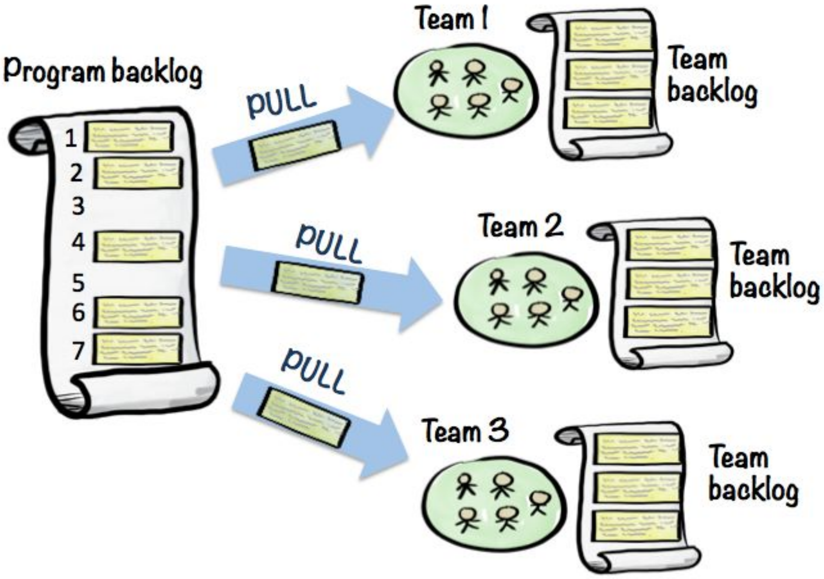
You notice the facilitators often reinforce this message: it is up to you, as an individual, to make the best use of your time during these days - whether it is planning with your team, chatting at the coffee machine, or discussing architecture. Don't wait for someone else to tell you exactly what to do or where to go.

However there are some constraints, so on each table you see a handout describing the overall planning process and the NFRs ("Non-functional requirements" - a pretty strange name if you think about it...). Things like legal requirements, service pack updates or new deployment procedures that all teams need to take into account. The NFRs are updated for every PI.

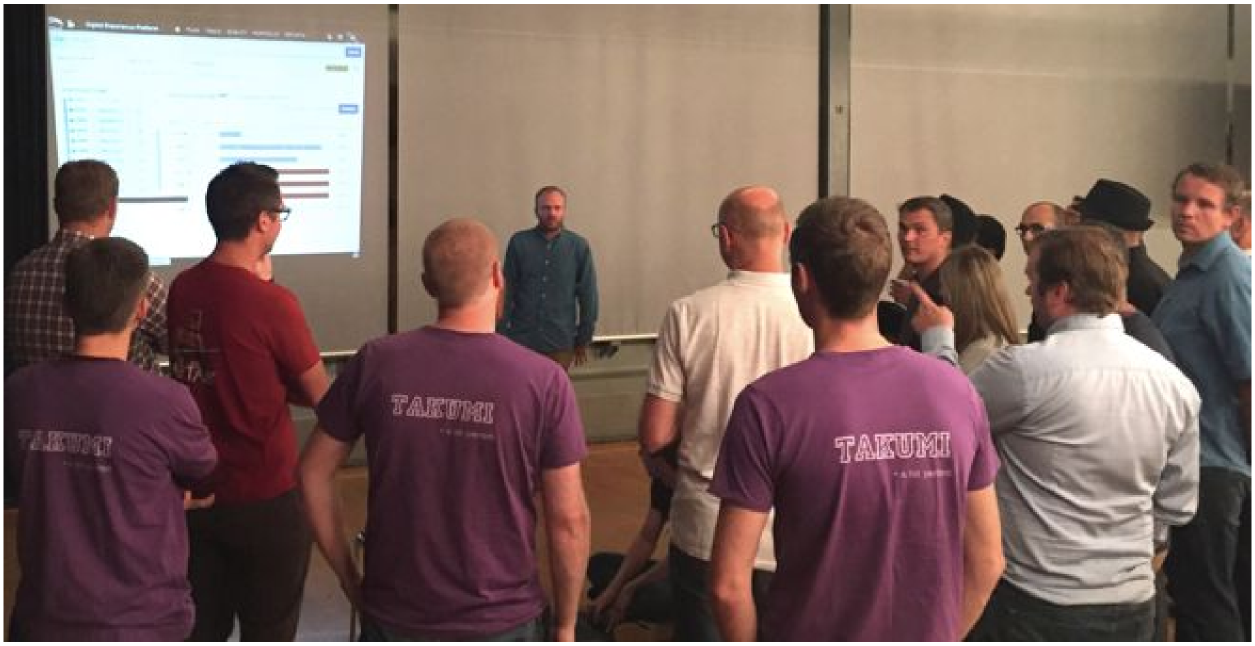
Pull-based planning

You notice an important detail - the teams are (for the most part) *pulling* in work, as opposed to having work *pushed* on them. They decide how much to pull in based on data about how much they've delivered in the past, and their gut feeling about what is realistic. To make this work in practice, each team has a product owner who negotiates with stakeholders and prepares a prioritized product backlog before the PI planning event. So product owners and clients own the priorities, while teams choose how much to pull in.

Some teams are fairly independent and have their own backlogs. Others, like the platform teams, work together and pull features from a shared program backlog.



You find the program backlog projected on a wall, and notice a mix of people from different teams standing in front of it, negotiating about which feature should be pulled by which team.



The teams are semi-specialized, each focusing on different areas such as LEGO ID authentication, cloud technology or search. The teams tend to pull features that are most familiar, or least icky, so the product owner sometimes needs to remind them that SOMEONE better pull that high-priority icky feature, even if it doesn't match your specialty. By now the teams have gotten pretty good at load-balancing as necessary to ensure that we, as a whole, are focusing on the right things (instead of super-efficiently building the wrong things). Team members are mostly T-shaped, which means they are somewhat specialized but also broad enough to be able to help each other.

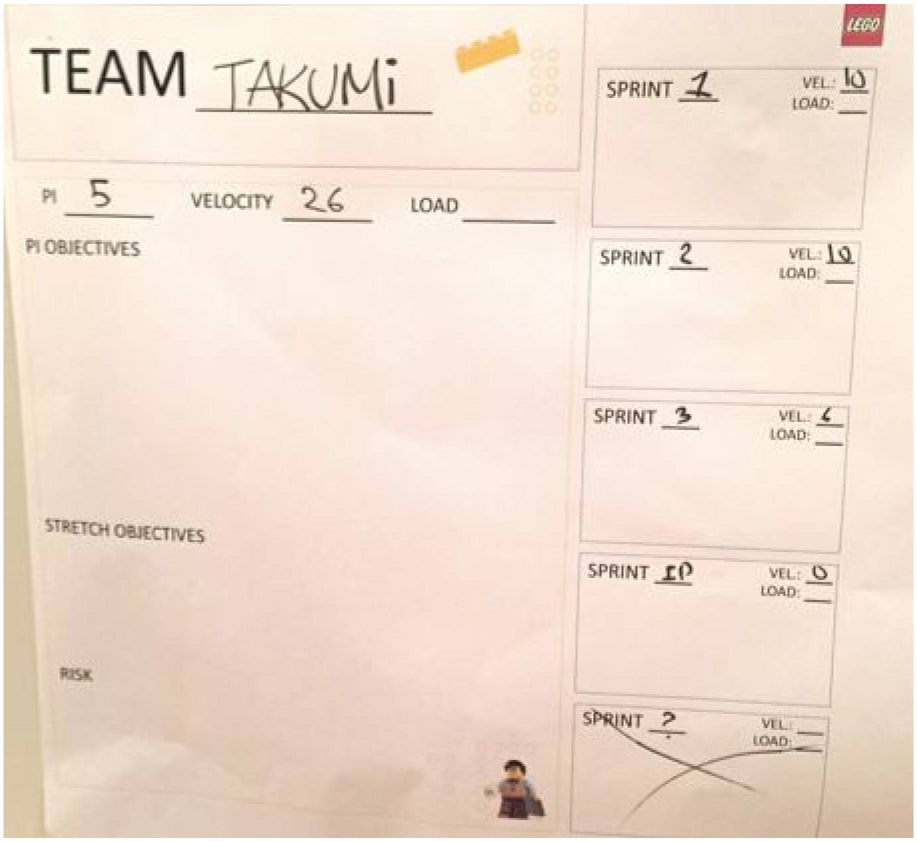
We used to put the program backlog on a physical board with printed cards, like this:

…. but that was kind of fiddly, so we figured out a way to do it directly online in our backlog management tool, projected on the wall. Less hands-on, but since we have like six teams doing platform development it gets easier to keep track of where everything ended up.

Looking around, you notice that some teams have big screens on their table, showing their backlog digitally, while others use mostly analog tools.

Team boards

Each team has a physical Team Board on a rolling whiteboard.


TEAM TAKUMi

PI 5 VELOCITY 26 LOAD ____

PI OBJECTIVES

STRETCH OBJECTIVES

RISK

SPRINT 1 VEL: 10 LOAD: __

SPRINT 2 VEL: 10 LOAD: __

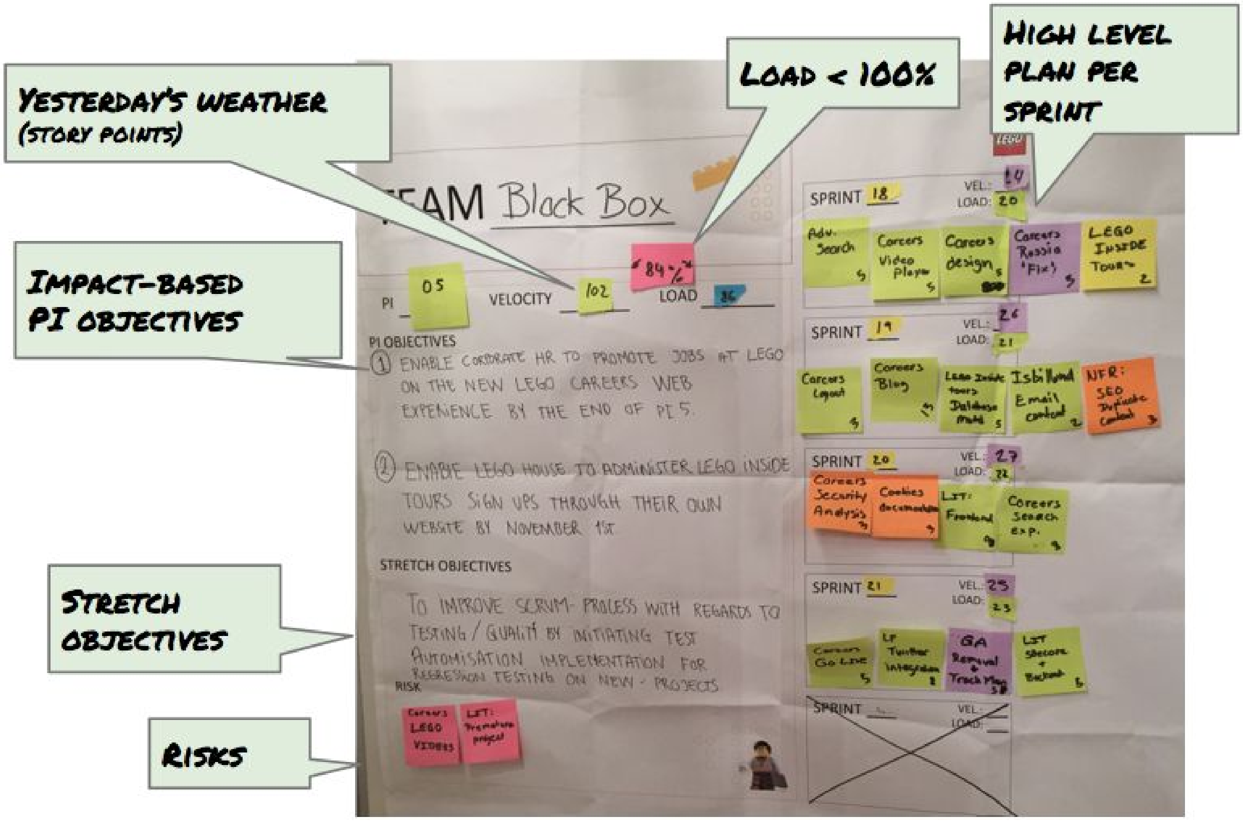
SPRINT 3 VEL: 6 LOAD: __

SPRINT IP VEL: 0 LOAD: __

~~SPRINT ? VEL: __ LOAD: __~~

It starts blank and gets populated during the day as the plan takes shape.

Here’s what the different sections mean:


Yesterday's weather (story points)
Load < 100%
High level plan per sprint
Impact-based PI objectives
Stretch objectives
Risks
TEAM Black Box
PI 05
VELOCITY 102
LOAD 86
84%
PI OBJECTIVES
① ENABLE CORPORATE HR TO PROMOTE JOBS AT LEGO ON THE NEW LEGO CAREERS WEB EXPERIENCE BY THE END OF PI 5
② ENABLE LEGO HOUSE TO ADMINISTER LEGO INSIDE TOURS SIGN UPS THROUGH THEIR OWN WEBSITE BY NOVEMBER 1ST
STRETCH OBJECTIVES
TO IMPROVE SCRUM-PROCESS WITH REGARDS TO TESTING / QUALITY BY INITIATING TEST AUTOMISATION IMPLEMENTATION FOR REGRESSION TESTING ON NEW - PROJECTS
RISK
SPRINT 18 VEL: 24 LOAD: 20
SPRINT 19 VEL: 26 LOAD: 21
SPRINT 20 VEL: 27 LOAD: 22
SPRINT 21 VEL: 25 LOAD: 23
SPRINT VEL: LOAD:

This is basically the team's high-level plan for the next 4 sprints (each sprint is 2 weeks, so a product increment is 8 weeks), the main output of the planning event.

- "uh… you guys release once every 8 weeks only?"
- "No, 8 weeks is just our program-level planning cycle. The release cycle is completely separate, some teams release more often, some release more seldom".

Of course the plan is going to change, depending on the volatility of each team's environment. But having a plan is still useful, because it drives all kinds of conversations around priorities, interdependencies, capacity, and so on. Demand will always exceed capacity, so the teams and clients need to work together to make the tough tradeoff decisions.

To aid the planning, the teams use Yesterday's Weather. That is, they look at data from past PIs, showing how much they got done (in story points). That little bit of data acts as a reality check to avoid overcommitment. We used to have a lot of problems with overcommitment, which in turn led to bad quality, missed deadlines, and distrust between client and team. Faced with uncertainty, it's better to undercommit and then pull in more work later, than to overcommit and then have to push out work later.

"PI objectives" are a high level commitment from the team, ideally impact-based ("we will achieve business result X") as opposed to output based ("we will deliver feature Y"). But our mileage varies. Stretch objectives are things that the team might finish, or hope to finish, but don’t feel confident enough to commit to.

- "But wait, what does “commitment” mean anyway?"

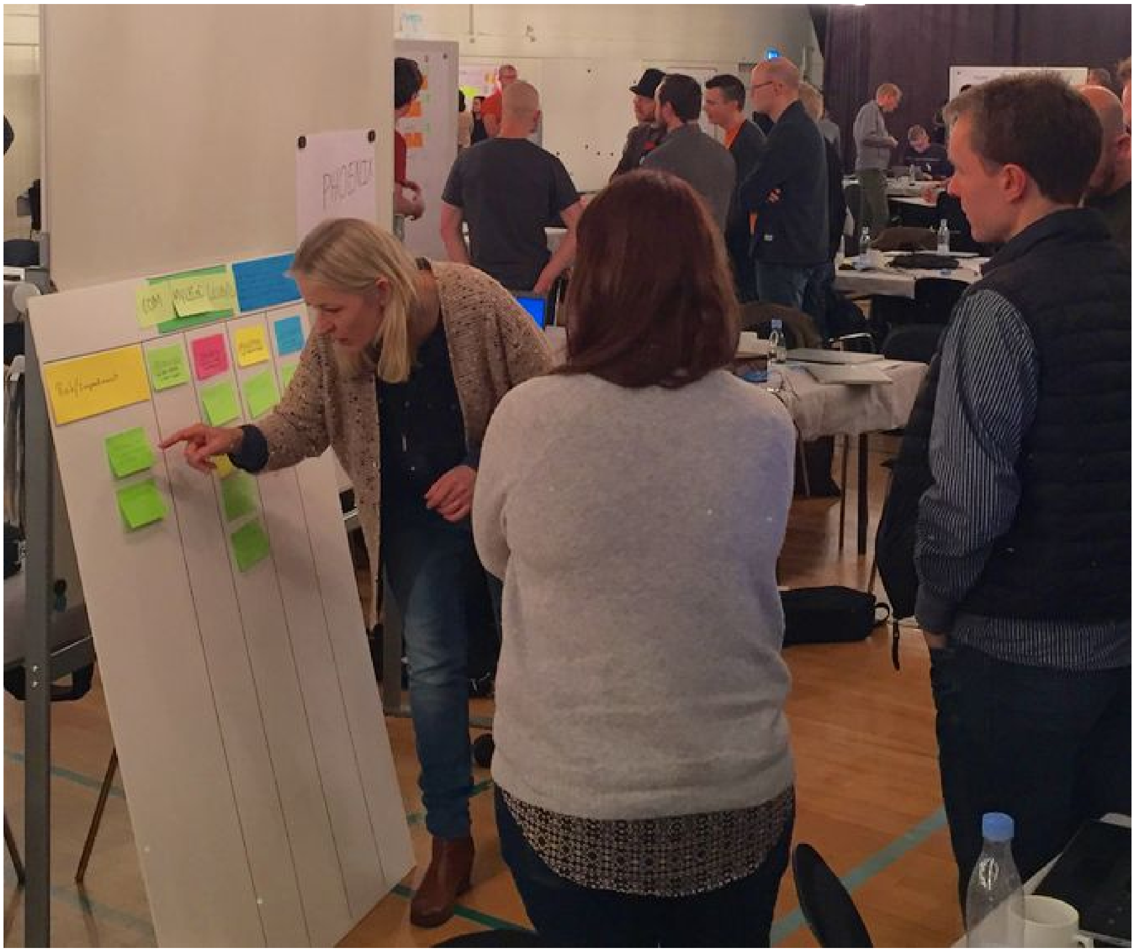
Glad you asked! A committed PI objective means:

- “Based on what we know right now, we honestly believe that we can fulfill this”.
- “We have spare capacity to deal with uncertainties”. How much spare capacity is needed? Depends on:
    - How uncertain are we about the amount of work involved?
    - How uncertain are we about our environment (changing priorities, etc)
    - How important is this commitment? The more important it is, the fewer other objectives we can commit to.
- “We will do our best to achieve the commitment, but we can’t be 100% sure.”
- “If we at any time stop believing that we can fulfill this, we will let the stakeholders know ASAP”.

In the past, commitment often meant “an unrealistic promise that someone made on your behalf. Deal with it”. That hurt quality as well as motivation, and made planning and forecasting really difficult.

Risk boards

As the planning progresses, teams start identifying risks that may cause the commitments to fail, potential problems like “the new licenses won’t be delivered on time”. These are posted on risk boards that are spread around the room, one board for each project or major initiative. Some risks get resolved locally by just talking to the right people, while others need to be escalated and stay on the board for the management review at the end of the first day.

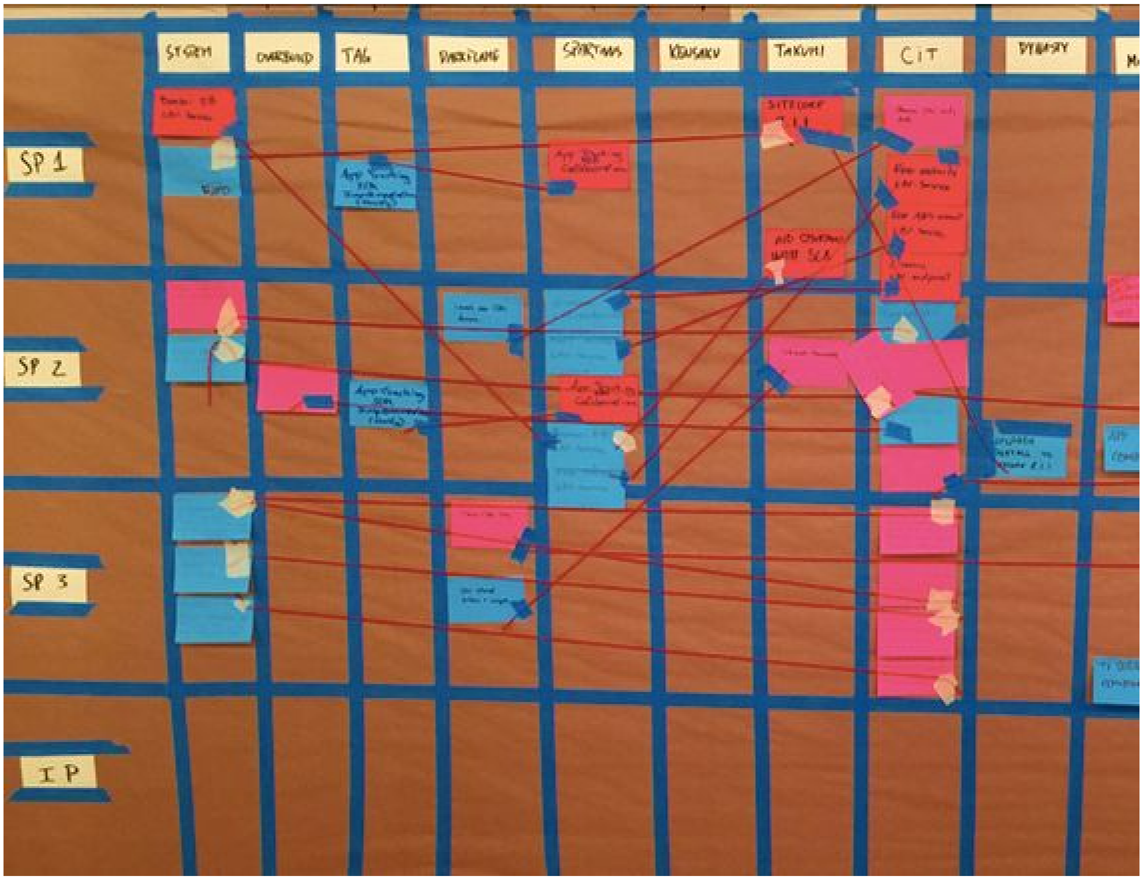


In the past, risks tended to be either ignored or all-too-quickly delegated to management (turning them into a bottleneck). By visualizing risks in a bottom-up fashion, teams have gotten a lot better at taking ownership themselves and only escalating risks they really need help with. Sometimes the cost of mitigating the risk is huge and out of proportion to the impact, so it's better to just accept the risk. Acknowledging and accepting the risk removes frustration and gives the team peace of mind, so they can focus on delivering instead of worrying.

Dependency board

Soon you notice a REALLY huge board hanging on one wall. It seems to be some kind of center of gravity, because there's always buzz and people in front of that board. It starts empty, but by the end of the first day it is a tangled mess of sticky notes connected to each other with… uh…. red yarn?! Stickers, scissors, yarn, what's up with THAT?

We call it the “dependency board” (sometimes “program board”). It shows who needs what from whom when. Take a closer look….



Each column is a team, each row is a sprint (2 weeks). The notes represent dependencies, from blue to pink. “In order to deliver THIS (blue sticky), we need THAT from you (pink sticky)”. You notice plenty of discussion and negotiation around what is going to be delivered to whom by which sprint.

Nobody seems to own the dependency board. The facilitators put it up initially, but then the teams completely self-organize around it, visualizing their dependencies to each other and using the board as a form of handshake protocol, to figure out who needs to talk to whom. The board is a centralized tool to enable decentralized collaboration.

- “So teams post all the features they are planning to deliver?”
- “No, only features with dependencies. We used to put *everything* the board, but that got too cluttered and nobody could make any sense of it. We concluded that what really matters is dependencies, so we focus on just that. Independent features are visible on the respective team boards, but not here.”

You spot an obvious bottleneck (the CIT column, Corporate IT), and animated discussion about how to utilize the bottleneck most effectively, and some conversations about how to reduce the bottleneck-factor over the long term.

- “What’s that all about?” you ask the facilitator.

- “Well, CIT is actually a separate organisation from DS, so initially they weren’t part of the planning process at all. But after the first couple of PI planning events it became evident that CIT was our most important dependency. People grumbled a lot about it. So we added a column for CIT on the dependency board, and started inviting representatives from that organization to join the PI planning events.”
- “Did it help?”
- “Yeah, made a huge difference! Became less blame game and more collaboration - less ‘us and them’, more ‘us and us’.”

CIT and DS people talk to each other face 2 face in front of the board and discuss how to make the best use of our precious bottleneck (and, of course, how to reduce the bottleneck factor over time). It looks pretty bad, but people at the board tell you “It used to be ALOT worse! You should see all the improvements we’ve made! Like moving things to the cloud to reduce dependencies. Takes time though.”

The bottom row of the dependency board is conspicuously empty. It’s called IP (“Innovation & Planning”). That sprint is held in reserve to leave space for unplanned innovation, overflow from the first three sprints, and other “stuff” such as the next PI planning meeting, training, and whatever else might come up.

An engineer points out “we’ve gotten a *lot* better at meeting our commitments now. We have capacity data (story points and such), so we are less likely to overcommit. But also, the IP sprint acts as a kind of buffer, so if something explodes we have space to recover. But most importantly, commitments are discussed and negotiated instead of just being dropped on our heads."

- “OK so what about innovation then? That’s the I in IP sprint right?"
  He laughs.
- “That part usually goes out the window. But it’s better now than in the early days. Last PI we did a hackathon during the PI sprint, lots of cool and useful stuff came out of that! Some team will always manage to carve out time for innovation during the IP, to the envy of the teams who couldn’t make it this time around.”

Another thing piques your curiosity:
- “What happens to the dependency board after the meeting?"
- “We roll it up, take it back to the office, and tape it up on the wall there."
- “And then what?"
- “Once or twice per week we do Scrum of Scrums. The scrum masters from each team gather at the board, discuss impediments and dependencies and mark them off as they get resolved."

He shows you a photo:

- “What about the teams in India, how do they access the board if it’s only on the wall in Billund?”
- “We’re still trying to figure that out...”
- “What about next PI planning? What happens to this board?"
- “Abandoned. At each PI planning we create a fresh new dependency board. Gives us a fresh perspective, and also lets us experiment with the format and structure of the board without being bogged down by history."

Draft plan fair

After a few hours of the seemingly chaotic breakout planning, the facilitator gets up on stage and announces that it is time for the draft plan fair. The buzz settles down as he summarizes the format (familiar to most people by now).

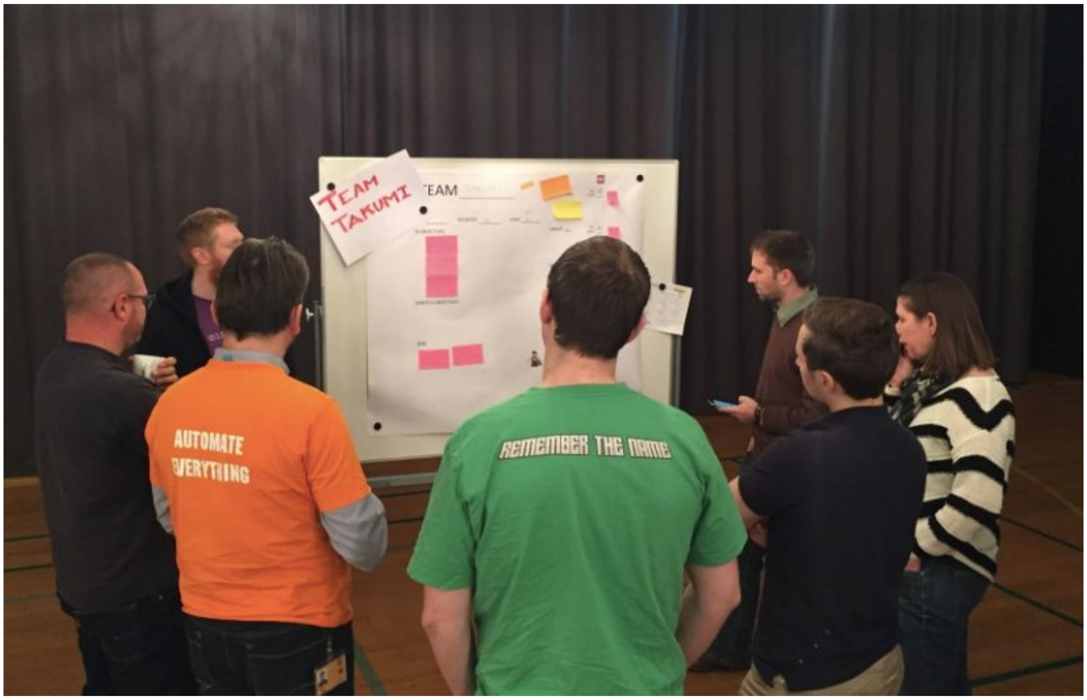
- “We’ll do four 7.5-minute sessions after each other. Each session, one team member stays at their team board to present the plan to others, while the rest can go see another team’s presentation."

He starts a big 7.5 minute timer and the presentations start. Small groups form around each planning board to hear and discuss the plan for that team - what they intend to deliver and why.

Dependencies and risks are discussed, and in some cases new problems are discovered. No panic, we'll have time to sort out the problems tomorrow.

- “Why 7.5 minutes? Why 4 sessions?”
- “We've experimented a lot with different timings, and this is what works best so far. Just enough time to learn something without getting bored or bogged down in detail.”

After 7.5 minutes, a bell rings and the second session starts. People move on to another team to hear their plan. And so on, four sessions in total, or 30 minutes.



- “OK, so each person can pick four teams to visit and learn about their plan?"
- “Exactly."

- “But what about the big picture? Isn’t it useful to hear all teams' plans?"
- “To a few people, yes. To the vast majority, no. So pick the four you care about the most. If you want to know more, you can visit more teams during the breakout tomorrow."
- “Is this the way you’ve always done it?”
- “No. In the past we used to do the draft plan sharing in a round-robin fashion. Everyone would sit down, and one team at a time would get up and present their plan to the whole room, while the facilitator points a camera to the board and streams it to the big screen. Pretty slick once we found the right tools. But even with strict timeboxing, it took at least an hour to get around all 20 or so teams.”
- “Ouch”
- “Yeah. Despite our best efforts to make it interesting, it was a total snoozefest! A few people listening attentively, but most people were staring vacantly at their phones or lying head-on-table waiting for it to Please Just End! People do care about what other teams are doing, especially teams they depend on. But that's like 3 or 4 other teams, they don’t need to know what ALL teams are doing. So we decided to stop force-feeding everyone with the plan, and instead do a pull-based system. The fair model worked MUCH better! “

You wander around, listening in on the presentation. People discuss their plans, dependencies, technlogy, risks, etc. Really does feel like some kind of fair!

The facilitator walks by again and points out:

- “Notice the high energy level in the room? That’s our main feedback loop, tells us we’ve got it right. If energy level is low and people start to zone out, no matter what we’re doing it’s wrong and we need to tweak the format. So we keep the parts that generate energy, and drop the parts that make people fall asleep. ”

You notice that there are actually 2 facilitators, working as a pair, taking turns on stage. He explains:

- "We always pair-facilitate this event. That way one can facilitate while the other takes a step back, observing the energy level and thinking about improvements. Pairing also provides some redundancy in case one of us gets sick, and makes it easier to onboard a new facilitator."

Management review and problem solving

At the end of the first day most people leave while the managers stay behind. Time for the management review!

The risk boards are gathered up in a row near the big dependency board, and the managers gather up in a half circle.

Everyone is affected by what's happening with the platform, so they start there. The platform product owner summarizes their plan and brings up a difficult tradeoff decision.

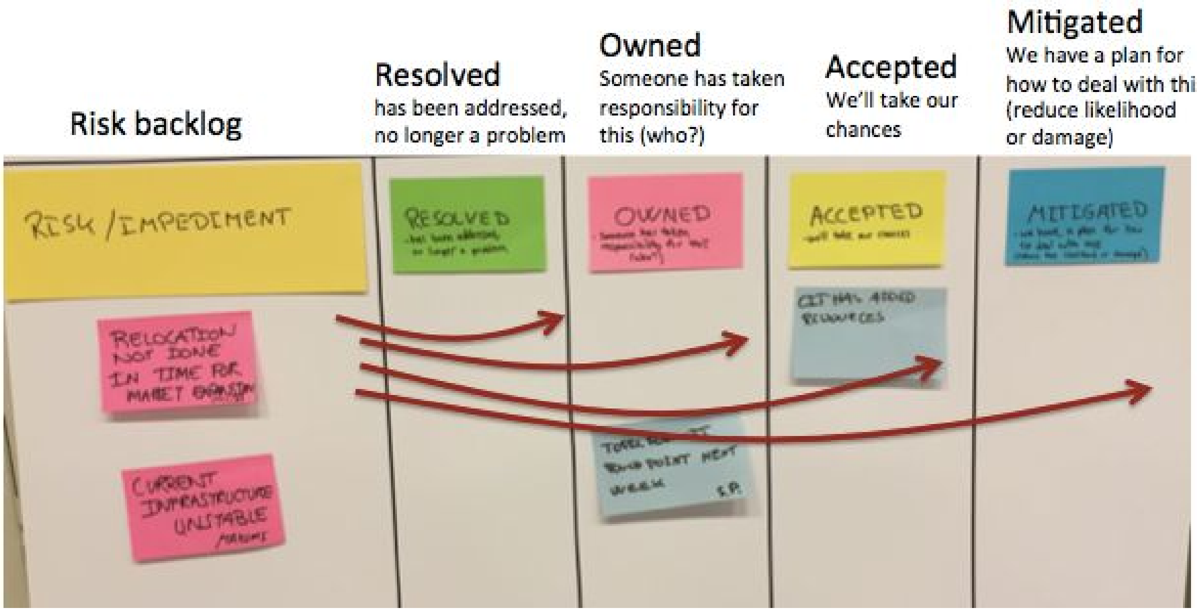
- "Bad news. We don't have capacity to deliver both A and B during this PI".

Both initiatives are really important and not doing one of them will have wider repercussions, so the debate is pretty heated. The facilitator helps keep things civil and focused. The managers discuss what options they have, pros and cons of each, and finally decide to prioritize A and postpone B. Several teams will need to adjust and realign their plans tomorrow based on this.

Next, they go through each risk board. The risks still on the board are the ones the teams couldn't resolve themselves, because they involve different parts of the organization or are outside the team's sphere of control. This group of managers now need to take ownership and decide what to do, and they are incentivized to do so because it's the only thing keeping them from going home...

You notice the columns on the risk board. One of the managers explains:

- "The ROAM framework is pretty useful for this conversation. We go through one risk at a time, discuss it, make a decision, and move it to one of the four columns: Resolved, Owned, Accepted or Mitigated ('ROAM the board'). So the go-home criteria is very clear. All risks on the board must be discussed and ROAM'ed!"


Risk backlog
Resolved
has been addressed, no longer a problem
Owned
Someone has taken responsibility for this (who?)
Accepted
We'll take our chances
Mitigated
We have a plan for how to deal with this (reduce likelihood or damage)
RISK / IMPEDIMENT
RESOLVED
OWNED
ACCEPTED
MITIGATED
RELOCATION NOT DONE IN TIME FOR MARKET EXPANSION
CURRENT INFRASTRUCTURE UNSTABLE
CIT HAS ADDED RESOURCES
TALK POINT NEXT WEEK
S.P.

After going through each risk, they agree on who will summarize this for everyone tomorrow morning.

You recall a book you've read on servant leadership, and realize that this whole setup kind of nudges managers into the servant leadership pattern. Teams see managers take responsibility and help solve problems, which builds trust.

The facilitator confirms this.

- "The management reviews used to be pretty chaotic and energy-draining. First we didn't have risk boards at all, so the conversations were pretty unstructured and took a long time, and decisions were sometimes lost or forgotten. Later when we introduced the risk board, there were just too many risks being escalated. Gradually the teams got better at taking ownership themselves (encouraged by the managers of course), and because fewer risks got escalated the managers were more able and willing to take ownership of those."

Day 2 - stabilizing the plan

The next morning you ask someone over breakfast:

- "Why a 2nd day? Aren't you guys done? Each team has a plan, dependencies identified, risks addressed, etc. Now what?"
- "The plan is just tentative. When we went home yesterday there were many unresolved problems. Now that we've slept on it, we can iterate on the plan and sort out problems that otherwise would show up downstream and mess up the sprints."
- "So you always do 2 days?"

- “So far, yes. But now we’re getting pretty good at it, and the 2nd day is pretty low-energy, so we’ll probably try a 1 day format some time.

  Fast forward: We did and, indeed, 1 day worked better! Go figure. More on that later.

After breakfast, the managers get up and stage and share their feedback about the draft plan that came out yesterday, summarize some key decisions from the management review, and explain how they will deal with the key risks that were escalated to them.

One big issue that comes up is unstable test environments, several teams had identified that as a big risk. The manager responsible for that area concludes that “we know about the problem, and most of the priorities this PI are centered on improving the stability. Talk to team X if you want more info. But it will take months to get a truly stable environment so until then we’ll just have to accept this risk”. Not great news, but the teams appreciate that the problem is acknowledged and that things are at least moving in the right direction. Some will adjust their plan to take into account continued instability.

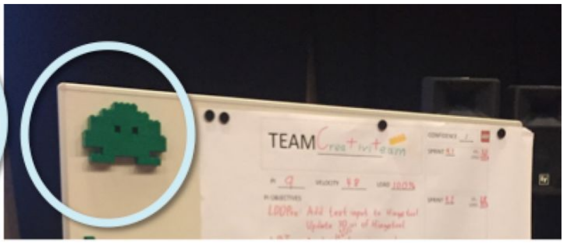
After that, the teams get back into breakout planning, same format as yesterday. Some finish early and start working on other things, while others have lots of dependencies and continue working with other teams to figure out what needs to be done by whom and when.

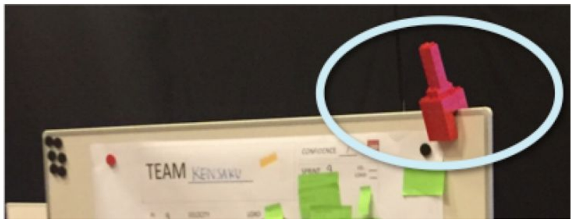
The facilitators encourage people to stay around even if their part of the plan is done. They can work on other things, write code, do email, whatever.

- “Sometimes a problem will pop up in another team that is dependent on you, and if you are still around the problem can be resolved quickly."

In some cases part of the team goes home while a few representatives stay around to be available for other teams.

By early afternoon the plans are stable (or as stable as they’ll get), and it’s time for the final plan review. Same format, 4 sessions where teams can go around and listen to each other’s plans. Colored LEGO bricks on each board are used to indicate whether the plan has changed a lot since yesterday, kind of like a status flag. Helps people decide where to go.





Confidence vote

After the draft plan fair, the teams go back to their table and the facilitator announces that it's time for the confidence vote. "How confident are you that you will be able to meet your PI objectives?". Each person holds up 1-5 fingers, where 1 means "forget it" and 5 means "totally confident".

Looking around, you see that most teams are pretty confident about meeting their commitments. However there are a few ones and twos so the facilitator asks them to say a few words about it. All in all the energy level is pretty low, so you can't help but wonder - what's the actual value in doing this?

Mini-retrospective

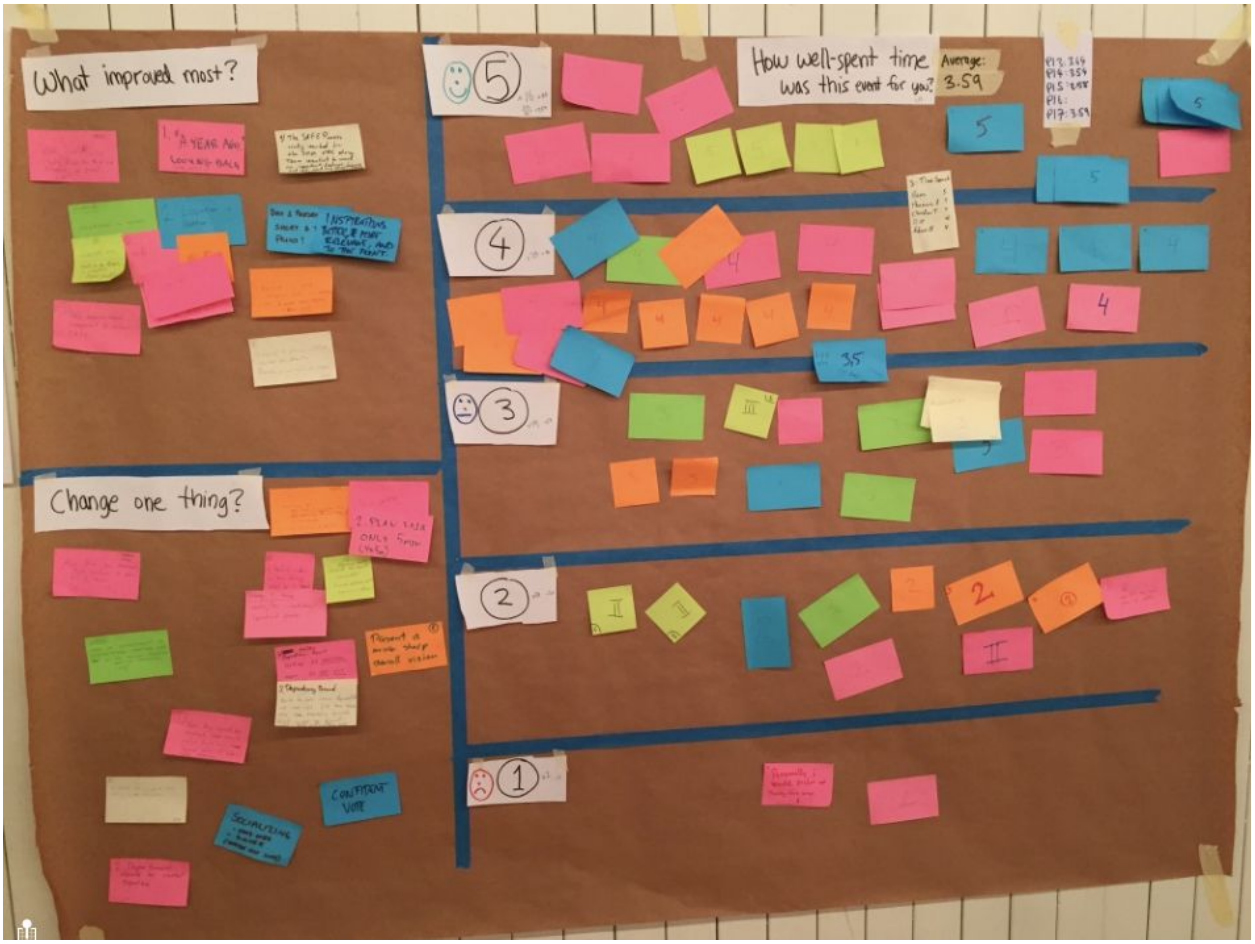
The last point on the agenda is the mini-retrospective. Each team sits down and talks about the planning event itself, what improved from last time, and what should be improved for next time. In addition, each team member individually and anonymously rates the value of the whole PI planning event ("how well spent time was this for me") on a 5-point scale.

Afterwards, everyone goes home except the scrum masters and the facilitators. They gather in a half-circle in front of an improvement board and, one by one, they post all individual ratings and summarize the feedback and improvement suggestions from their teams.

After everyone has had their say, they take a step back and look at the board, identifying patterns and themes. For the most part people liked the event (mostly 4s and 5s) and considered it to be well-spent time.

- "Wait! Aren't these mostly developers? I thought developers HATE meetings?"
- "Yeah, we were worried about that when we started with these planning events. But the ratings were higher than we expected, even the first time. Seems like people see this as actual *work* more than a meeting."

There are some 2s though, and even a couple of 1s, but most of those people have actionable feedback on how to improve the event next time. They like the idea of doing big room planning, and their concerns are more about how we execute them. Or the length.



One thing that comes up at the mini-retro this time is the confidence vote. How useful is that, actually? Some want to tweak it, others think it is waste and should be removed. After some discussion they agree on an experiment - “let’s skip confidence voting entirely next PI planning, and see if we miss it!”

> *Fast forward: So we skipped confidence voting next PI and, as suspected, we didn’t miss it! The team commits to their plans based on pull rather than push, which by definition means they are confident enough, thus making the vote redundant. Bye bye confidence voting! That’s just one of many examples of how we keep experimenting and tweaking the format to maximize value and minimize waste.*
>
> *Fast forward even further: In Q3 2016 we tried a 1-day version (same activities, but squeezed into 1 day instead of 2), and saw a clear improvement in energy level as well as the rating! So we stayed with the 1 day version. But general consensus is that we*

> *probably needed the 2 days in the beginning, as a stepping stone until we could learn how to do it more effectively.*

And that's it. Done. The teams take their team boards and risk boards and go home. The facilitators stay around to do a short debrief with each other, jot down some notes about what to do differently next time, and then starting packing up the dependency board and other things.

There, now you've seen a PI planning! Hope you enjoyed it.

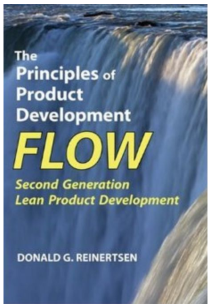
Now let's take a step back.

What else changed, other than big-room planning?

Lots! Although the PI planning is the center of gravity for the changes we made, there is obviously more to it than that. The 'real' work is really what happens in between PI plannings.

During the PI, teams are generally doing Scrum and 2-week sprints, with the usual Scrum ceremonies such as sprint planning, daily standups, backlog refinement, sprint review, and sprint retrospective. We also have a shared PI demo and retrospective. But we won't drown you in details about that here. This article focuses on the PI planning event, since that illustrates the bottom up, decentralized coordination that we are trying to achieve. The PI planning is like a looking glass into the overall process and culture.

One activity worth emphasizing is the PI pre-planning, where product owners get together to discuss and prioritize features for the upcoming PI. We have three such pre-planning sessions before each PI-planning. There the features evolve from just a headline to well understood features prioritized by value, time criticality and effort. Read up on Cost of Delay or WSJF (Weighted Shortest Job First) if you want to learn more. If you want to go really deep, check out Don Reinertsen's book "Principles of Product Development Flow".



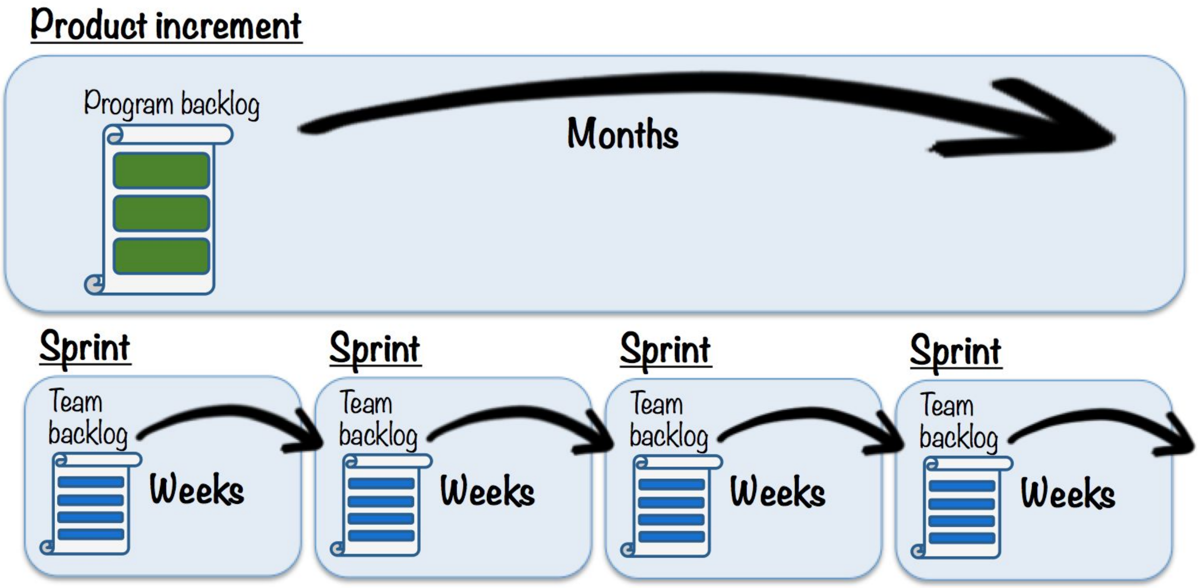
What's the actual purpose of the PI planning event?

In one word: Alignment

PI planning is a big room planning event, and the purpose is to get teams aligned with each other.

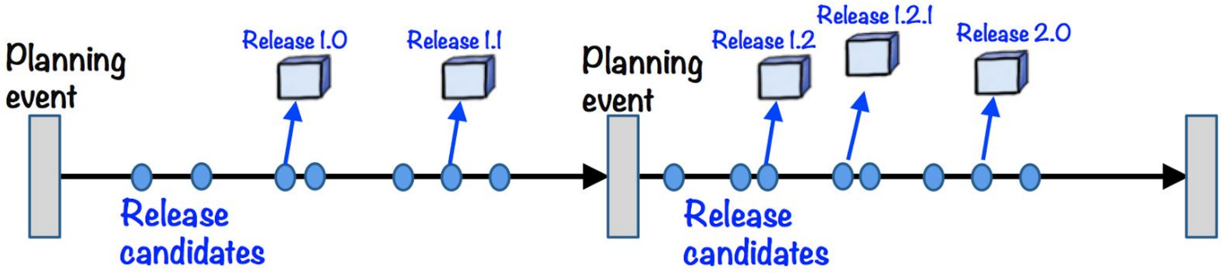
The value of big room planning is directly related to how many dependencies you have. The best is of course to design your team structure and architecture to minimize dependencies and avoid the need for things like big room planning. But if you have a bunch of teams working on the same product, or (like in our case) closely related products with shared technology, then big room planning is super-useful.

With sprints, each team has a 2-week planning horizon. PI planning adds a higher level planning horizon, where teams get together and look further into the future (2 months, one “product increment”) but with less detail.

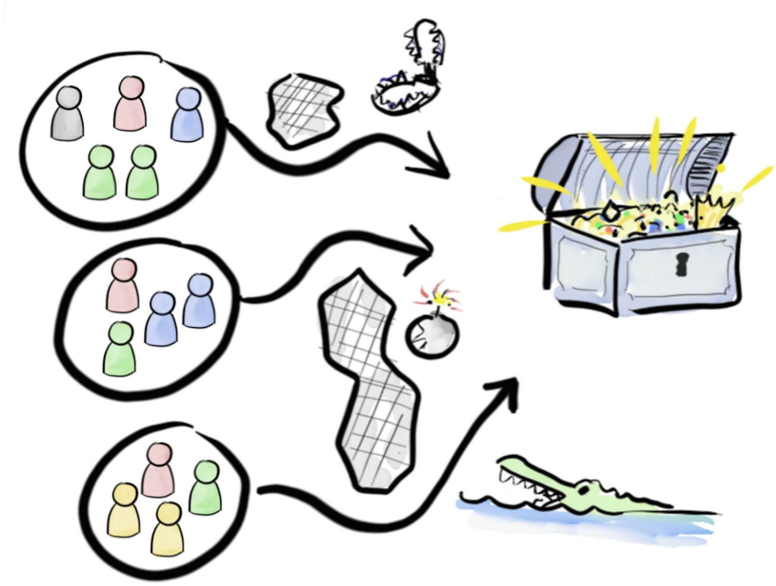


The trick is to avoid the temptation of creating precise, detailed plans. Better to be roughly right rather than precisely wrong.

Note that product increment, despite what it sounds like, isn’t a big bang release. It’s a synchronization point, a time and place where teams get together to align with each other. This is completely decoupled from the release cycle. Different teams release on different cycles, depending on the nature of their work. So we plan on a cadence, and release on demand.



The PI-planning solves the problem of misaligned teams and all the frustration and waste that creates. It is a platform for giving people a bearing towards a shared goal, a “Big Heart Beat” that synchronizes a bunch of more or less autonomous teams.



Nothing beats face to face. The sheer amount of information being shared and the number of decisions being made during PI planning is actually quite astonishing. This is possible because all brain trust and empowerment is summoned in the same room at the same time. Anyone you need to talk to is right here in the room, right now. No need to send a calendar invite or wait for a suitable meeting time, just go talk.

Without the planning event, we would need a bunch of separate coordination meetings and emails and spreadsheets to accomplish the same thing, which adds waiting time and misunderstandings. The finer details of the work is discussed after the PI planning (during the sprints), but the skeleton is created at the big room planning event.

Another benefit of the PI planning is that it creates radical transparency, revealing the inherent complexity of large scale software development. This can be a little intimidating to outside stakeholders, but they gain a better understanding and increased respect for the teams, for their work in actually trying to tame the complexity monster.

So what's the downside then? Well, it's tempting to say cost, but that's actually not such a big issue. People are paid the same salary whether they are at a PI planning or not. The only real overhead cost is the venue and food, which is a small price to pay for the value gained.

One psychological downside is that, if you are a control freak, you will not come out with peace of mind and a crystal clear plan. The whole setup can feel pretty chaotic and unstructured.

Once you get used to the format, however, you realize that the meeting provides a good enough overview to make you comfortable that the objectives will be met, while still allowing you to zoom in on the areas that you find particularly important. The rest you need to leave to the teams in good faith - after all, they are generally pretty smart and dedicated people, why else hire them :)

What impact has this given?

Nothing is perfect, but overall the impact has been surprisingly positive, and nobody seems to want to go back.

- **Less duplicated work**. Teams are more in tune with each other, so they waste less time on redundant work.
- **Less dependency problems.** Teams waste less time being blocked waiting for each other. Teams interact more smoothly with other departments and stakeholders.
- **Managers can update priorities and resolve impediments faster**, because they have a better idea of what is actually going on.
- **Client trust has improved**, because they have a better understanding of what the teams are working on and why.
- **Planning is easier and commitments are met more often**, because the teams and portfolio planners learn how much work we can commit to and what our actual capacity is.

More importantly, this has **improved the motivation** of the team members. Going to work is more fun when there's less confusion and less waste. And motivated people do better work, so it's a positive cycle!

Another impact we've seen is that other parts of LEGO visit the meeting, get super inspired, and start exploring how to implement some of these principles and practices in their own department. In fact, agile is spreading like a virus within the company, and the highly visible nature of the PI planning event is like a a catalyst.

Nevertheless it’s still complex and hard work (“hard fun” as we like to call it) and there is lots of room for improvement. But by becoming more aligned towards shared goals, getting more empowered teams, getting better at setting the right expectations for our clients and better at identifying interdependencies - all this gives us the sense that we’ve gotten better at delivering the right things in the right way.

How did all this get started?

You still here? Wanna know how this all got started in the first place? OK, here’s the back story.

In the late nineties life was good. A handful of people (cross functional team we would call it today) would craft all of LEGO’s online presence with a few GIF’s and HTML. But as digital grew, so did the number of people required to deliver digital solutions, and the development tools became much more advanced. At around 15 teams we were really stepping on each other’s toes and pulling in different directions.

At first we tried to throw a bunch of project managers at the problem, but it didn’t really help. They always seemed to be a step behind. Typical symptom: constantly updating plans and asking steering committees for added funding or postponed launch dates.

We still believed in the agile principles that we had introduced on a team level around 2009, but we didn’t really know how to scale it. One of the project managers had heard of this thing called “Scaled Agile Framework” and he went to learn more about it. He shared it when he got back, and then we slept on it….for a couple of years.

But then suddenly the ball started rolling. After discussing with some of the teams, the project managers were confident that this framework could help us. So all the managers in the department were invited to an introduction to the framework. They weren’t entirely convinced, but nevertheless it seemed like it MIGHT help, and it seemed very unlikely to make things worse, so heck why not give it a shot? And so a senior manager backed it up and secured funding for training all 120 people in the department.

We had three days of training including a simulation, and the following day we made a leap of faith, created a program backlog and scheduled our first PI planning. It was a noisy, cramped, and chaotic start. But, to our surprise, we hit the ground running and the event worked out better than we had feared.

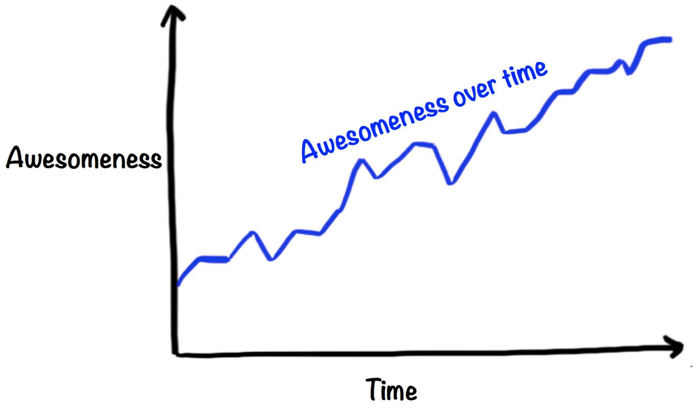
Our first PI planning – learning by doing

Here are some of the things that helped us get a successful start:

- **The teams *wanted* it to work.**
  If the teams did not buy into it we would have failed miserably. Despite scepticism about some parts of the framework, they really played ball and did their best to make it a success.
- **Existing agile experience.**
  The teams had been running agile for years, and a good part of the organisation understood the principles. This was a really good foundation to scale up from.
- **Management buy-in.**
  There was enough support and faith from management to allow it to succeed. Although quite few senior managers were directly involved they accepted the risk involved, gave room for it to unfold and made sure we had enough financial support to educate and train everybody involved.
- **Non-dogmatic approach.**
  Although we used the Scaled Agile Framework as a launchpad, we didn’t follow the framework religiously. The framework contains *way* too much detail for our needs, plus it is optimized for a bunch of teams working on *one* product, while our teams work on a number of different products and services. So for each new PI we customized and tweaked the process, adding elements that we needed, removing elements that weren’t adding value. That gave people a sense of ownership, they could see that the process and structure is there to serve them, not vice versa.

Now what? What are your current challenges?

We keep experimenting. Each change either improves things, makes them worse, or makes no difference. But we keep the stuff that works and ditch the stuff that doesn't, so over time things gradually improve.



The biggest improvements were in the beginning, now the changes are smaller and we seem to have found a sustainable flow. Mostly minor tweaks and tradeoffs. With so many people in the room we can't really make sure everyone is super happy, but at least we can make sure no one is super unhappy.

Our biggest challenge is that we work on a bunch of different types of things. Although we are one department using mostly the same tools and technology, we are not building a single coherent product. Ideally we'd like to reorganize into a number of smaller clusters of teams based on specific products or value streams - our Grand Unified Theory of organizational design. Then could have a number of smaller PI plannings instead of the big monster meeting. But we haven't figured it out yet, and we've gotten used to the convenience of having everyone in the same room for the planning.

As a result, it is hard for us to focus the PI plannings on just one or two themes with a clear storyline. Instead we end up with a bouquet of different objectives catering to different teams and stakeholders. Unfulfilling for managers, because they want a clear storyline of what the teams will be working on. And unfulfilling for the teams, because they want a clear understanding of current context and overall priorities. It's kind of a chicken and egg problem.

Although experimenting and tweaking the process seems like a good thing, we sometimes ask ourselves the meta-question: Are we really optimizing the right variable? Or are we changing things just for the sake of change? For example, if PI planning isn't the biggest constraint any more, should we really keep trying to improve it? So far we honestly believe we are improving the right things, but it’s good to check once in a while.

Another challenge is momentum. In the beginning everyone was excited (or anxious) about the changes, energy was high and the vibe was good. But sometimes we sink into a comfortable “business as usual” mindset and lose momentum. Small changes and surprises during PI planning can be a good thing to keep people thinking. Things can ALWAYS improve, we just need to remind ourselves about that from time to time.

Wrapup

Hope this article was useful to you! It certainly was useful to us - gave us an excuse to really reflect on what we’ve been doing and learning :o)

Just keep in mind, our approach isn’t the “answer” to agile scaling. It is just an example, a snapshot of a journey in progress. Agile is all about continuous improvement - it is a direction, not a place.

One thing is for sure though. If we had tried to plan out this whole change from the beginning, we would have gotten nowhere! Instead, if you want to make change happen, just start from where you are now, set some kind of a goal, and then start experimenting. It’s all about the people - if they are onboard with the change, if they support the idea, then you’ll make progress. Otherwise not.

And don’t be ashamed to use an existing process framework, or borrow ideas from other companies or case studies (like this one). No need to reinvent the wheel! Just make sure you adapt things to your context. You won’t find a complete solution that fits your needs, but you can probably find something that’ll get the ball rolling (or the bricks flying) in the right direction.

Good luck!
- Henrik & Eik

> *Acknowledgements: A lot of very smart, passionate, and friendly people have been involved in shaping these changes! We won’t bother trying to list everyone. Our role has been mostly to support the journey and fan the flames :o)*

”Dr Agile”

crisp

”Captain Agile”

LEGO

Henrik Kniberg
henrik.kniberg@crisp.se
http://www.crisp.se/henrik.kniberg
https://twitter.com/henrikkniberg
http://blog.crisp.se/author/henrikkniberg

Eik Thyrsted Brandsgård
eik.thyrsted.brandsgaard@lego.com

Feedback? Questions?

Use [blog comments]() or reach out to us directly. We’ll read most stuff but can’t guarantee a response unfortunately. Life is too short to always have to respond to everything :o)