

Respect People Principle of Lean means: Move responsibility and decision making to the lowest possible level

In which category do you belong?

Here is the litmus test?

When you are annoyed with your job, do you complain, ignore or fix it?

People come first

Value people. Trust them. Support them.
These values are the essence of building an Agile ecosystem.

Estimate is not Commitment

8/25/2023 8:26 PM

vijaynathani.github.io

Page 2

If a team member is repeatedly harming the team by their actions, then they are not acting in a professional manner. They aren't helping move the team toward a solution. In that case, they need to be removed from this team.

If the majority of the team (and especially the lead developers) don't act in a professional manner and aren't interested in moving in that direction, then you should remove yourself from the team and seek success elsewhere.

If estimate is regarded as commitment,

Programmers will add "safety time" just in case things go wrong.

Even if, later on, things go well, the estimated time will be used up, since estimators don't want to look that they over-estimated.

If estimates are not regarded as commitments

Work gets done faster overall.

Taylor was wrong:

Taylor believed that most fixed wage workers spent their time trying to figure out how to work slowly, since being efficient

brought no extra pay and
could threaten their jobs

His approach was to divide the assembly line work into very small steps and time the workers to uncover the “one best way” to do each step

Work on the assembly line was boring repetitive and tightly controlled.

The workers were shown exactly how to do their job and told how much time they had to complete it.

They could be trained in 10 minutes and replaced in ten minutes.

Productivity by Estimation

Effort estimate prepared by	Average Productivity	Number of Projects
Programmer alone	8.0	19
Manager alone	6.6	23
Manager & programmer	7.8	16
System Analyst	9.5	21
(No estimation)	12.0	24

Ref: Peopleware

8/25/2023 8:26 PM

vijaynathani.github.io

Page 3

If an approximate estimate is needed, the person who is going to do the work should give the estimate.

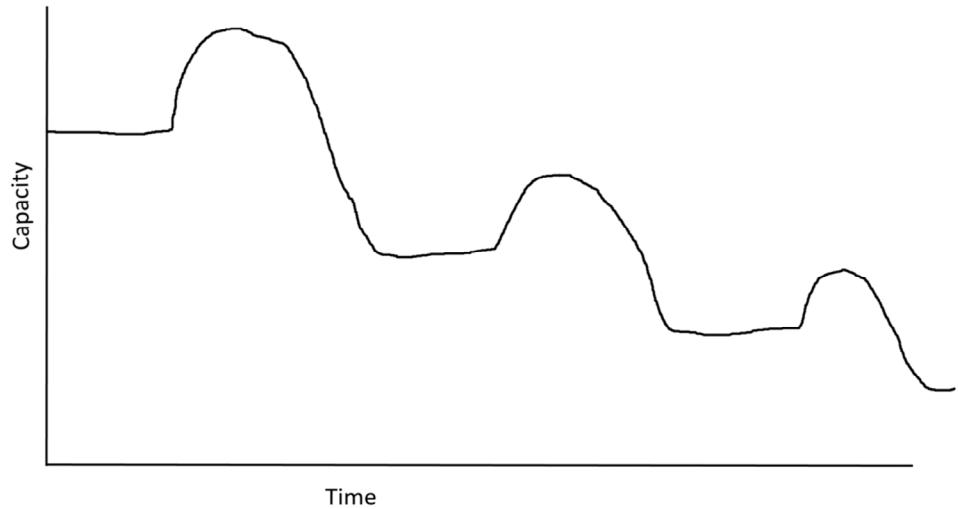
Parkinson was also wrong

Parkinson's law, "work expands to fill the time allocated for it."

People's lives are just too short for any loafing on the job. Since they enjoy their work, they are disinclined to let it drag on forever—that would just delay the satisfaction they all hanker for.

They are as eager as you are to get the job done, provided only that they don't have to compromise their standard of quality.

Capacity Trap

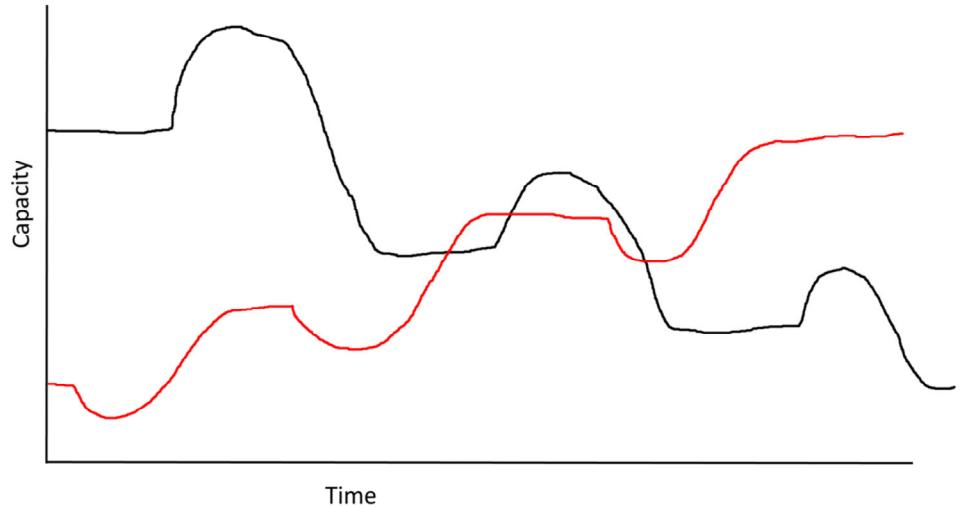


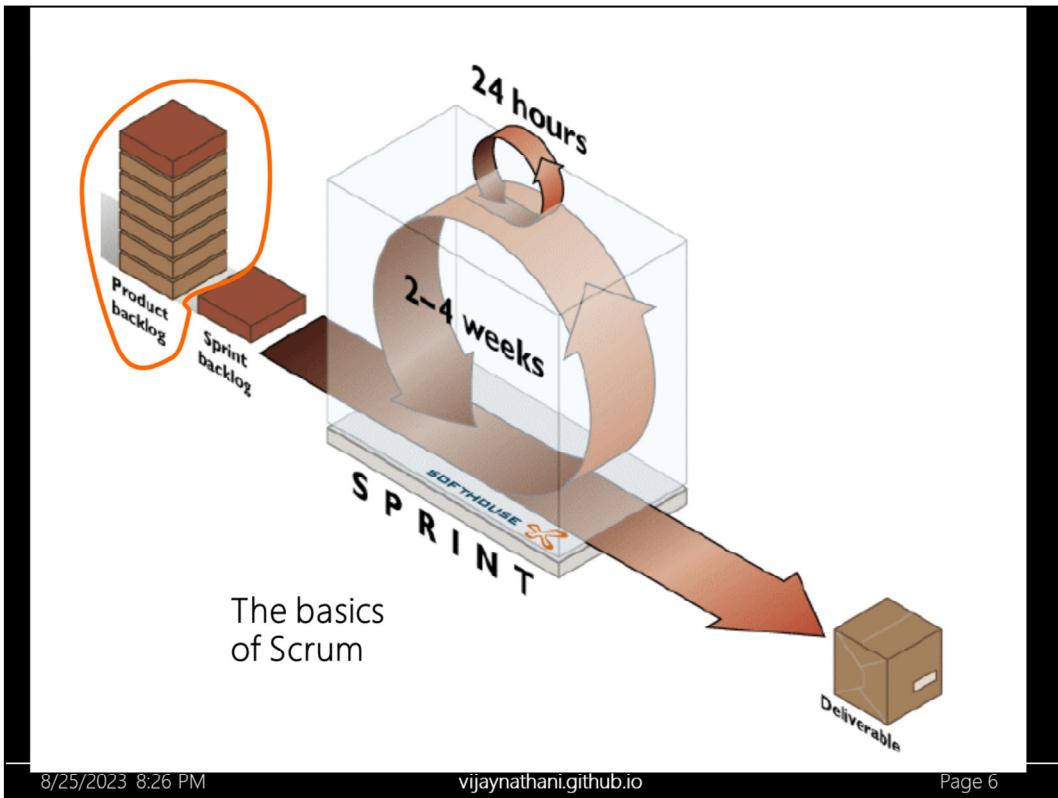
8/25/2023 8:26 PM

vijaynathani.github.io

Page 4

Capacity Trap - Solution





8/25/2023 8:26 PM

vijaynathani.github.io

Page 6

No Estimates

- How many user stories this sprint?
 - Answer: Same as last sprint.

Estimation is a waste of time. Let the project continue as long as the output is profitable.

<https://vimeo.com/user22258446/review/126266148/4543b8ba38>

<http://noestimatesbook.com/>

Product Backlog

Description	Rough Est. Size
Enable all users to place book in shopping cart (mocks and additional details are located here)	20
Upgrade transaction processing module (must be able to support minimum 500 transactions per second)	10
Investigate solutions for speeding up credit card validation (see target performance metrics located here)	20
Upgrade all servers to Apache 2.2.3	40
Diagnose and fix the order processing script errors (bugzilla ID 14823)	20
Enable all users to create / save wishlist	40
Enable all users to add and delete items on their wishlist	10

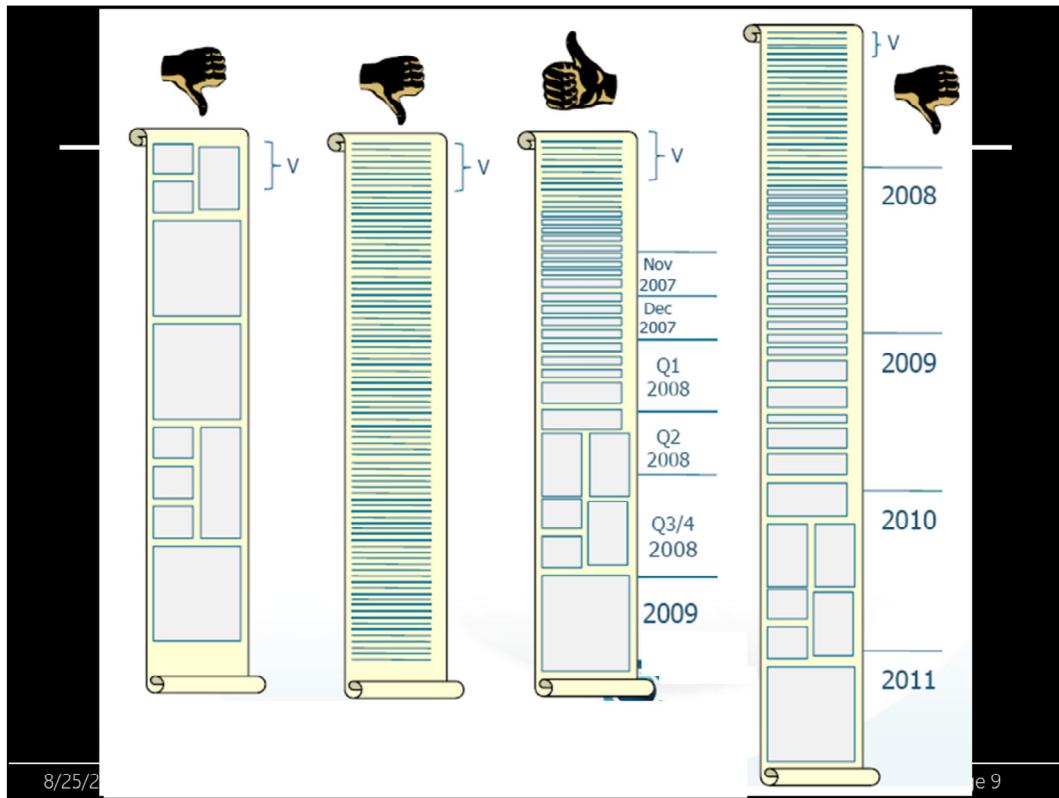
Product Owner lists items in descending order of priority (highest priority item is listed first, next-highest is second, etc.)



8/25/2023 8:26 PM

vijaynathani.github.io

Page 8



Product Backlog

List of everything that could ever be of value to the business for the team to produce

Ranked in order of priority

Priority is a function of ROI and risk

Product Owner can make any changes they want before the start of a Sprint Planning Meeting

Items added, changed, removed, reordered

The farther down the list, the bigger and less defined the items become

~2 Sprints worth are defined in detail

The Basics of Scrum



Sprint Planning Meeting

- Goal of Sprint Planning Meeting:
For the team to make a good commitment around what it will deliver by the end of the Sprint
- What's a good commitment?
 - Clearly understood by all
 - Shared among the team
 - Achievable without sacrificing quality
 - Achievable without sacrificing sustainable pace
- Attended by Team, Product Owner, ScrumMaster, Stakeholders
- May require 1-2 hours for each week of Sprint duration

Sprint Planning Meeting

3 Steps:

- 1 Product Owner, Team, and other Stakeholders talk through Product Backlog Items and prioritization.
- 2 Team determines how much time it has available to commit during the Sprint
- 3 Team selects as much of the Product Backlog as it can commit to deliver by the end of the Sprint, and turns it into a plan

8/25/2023 8:27 PM

vijaynathani.github.io

Page 12

Team:

Validates commitment by breaking down into tasks with time estimates

Team decides who will do what, when; thinks through sequencing, dependencies, possible task trades, and so forth.

The Product Owner answers questions but does not direct the team's choices.

The outcome is the Sprint Backlog.

Product Backlog

Description	Rough Est. Size
Enable all users to place book in shopping cart (mocks and additional details are located here)	20
Upgrade transaction processing module (must be able to support minimum 500 transactions per second)	10
Investigate solutions for speeding up credit card validation (see target performance metrics located here)	20
Upgrade all servers to Apache 2.2.3	40
Diagnose and fix the order processing script errors (bugzilla ID 14823)	20
Enable all users to create / save wishlist	40
Enable all users to add and delete items on their wishlist	10

8/25/2023 8:27 PM

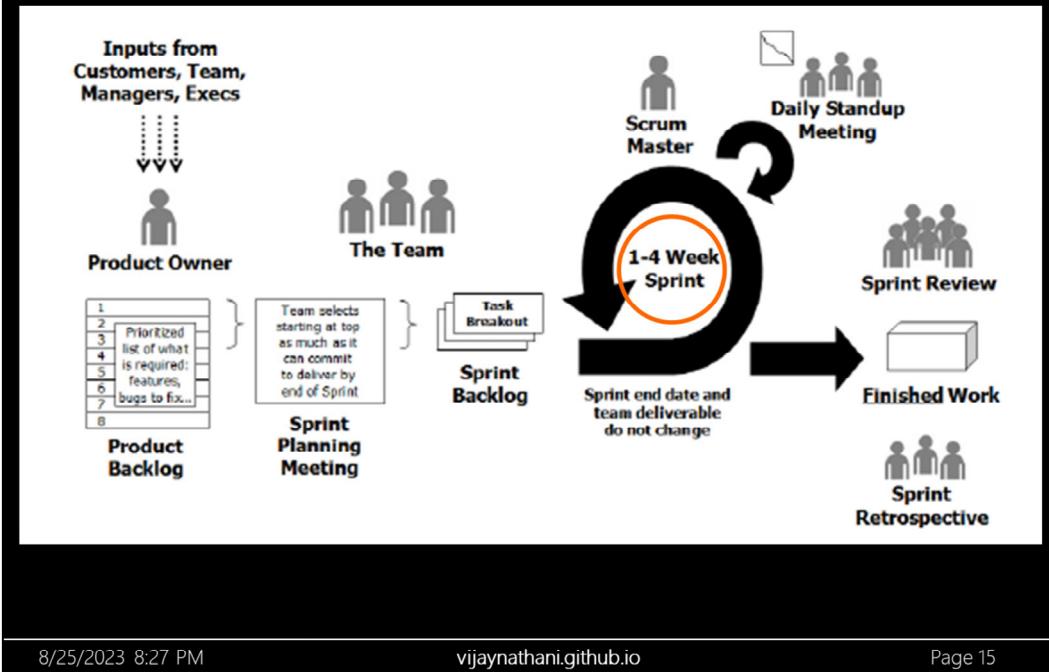
vijaynathani.github.io

Page 13

Task Breakdown

Backlog Item	Task	Owner	Initial Time Estimate
Enable all users to place book in shopping cart	Configure database and space IDs for Trac	Sanjay	4 hours
	Use test data to tune the learning and action model	Jing	2 hours
	Setup a cart server code to run as apache server	Philip	2 hours
	Implement pre-Login Handler	Tracy	6 hours
Upgrade transaction processing module (must be able to support 500 transactions /sec)	Merge DCP code and complete layer-level tests	Jing	8 hours
	Complete machine order for pRank	Jing	4 hours
	Change DCP and reader to use pRank http API	Tracy	2 hours

The basics of Scrum



8/25/2023 8:27 PM

vijaynathani.github.io

Page 15

Sprint Commitment

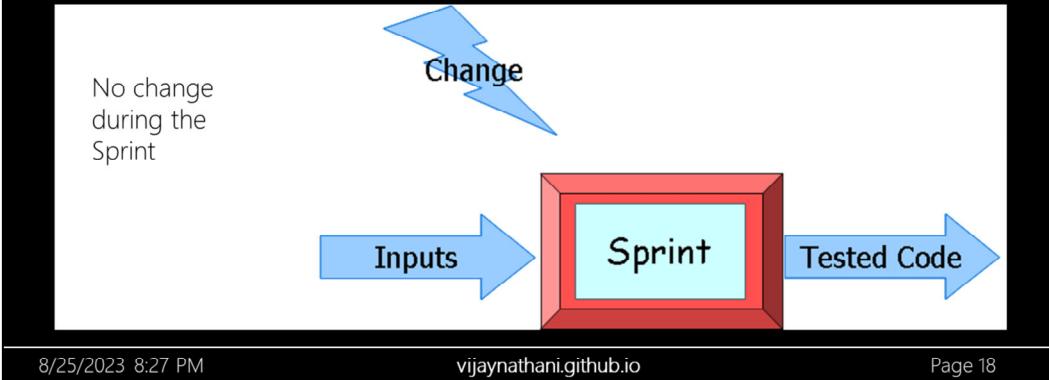
- Team's commitment to the Product Owner: We promise that
 - We believe that we can complete all stories included in the Sprint backlog.
 - We will do everything in our power to reach the Sprint goal and will let you know immediately if we can no longer reach it.
 - We will release for production at the end of the sprint.
 - We will display our progress and status on a daily basis.
 - Every story, what we deliver, is "Done".

Common Misconceptions

- We promise we will achieve the goal.
- We promise that we will deliver all stories included in the sprint backlog.
- Estimates are given by the Product Owner / ScrumMaster

Sprint

- Target duration is one week to one month
- Product is designed, coded, and tested during the sprint



The Sprint

The Sprint is never extended

Sprint ends on pre-determined date whether or not team has completed what it committed to

Sprint Length

4 weeks is standard in literature

2 weeks is also common

1 week is minimum, 4 weeks is maximum

Factors in deciding your Sprint length

Length of the release

Amount of uncertainty

How long priorities can stay unchanged

Overhead of Sprint Planning and Review

Urgency / intensity lag

Sprint Cycle: 2-Week Sprint

Mon	Tues	Weds	Thurs	Fri
		1	2	3
6 Sprint Planning Meeting	7 1	8 2	9 3	10 4
13 5	14 6	15 7	16 8	17 Sprint Review & Retrospective
20	21	22	23	24
27	28	29	30	31

8/25/2023 8:27 PM

vijaynathani.github.io

Page 19

Sprint Cycle: 2-Week Sprint

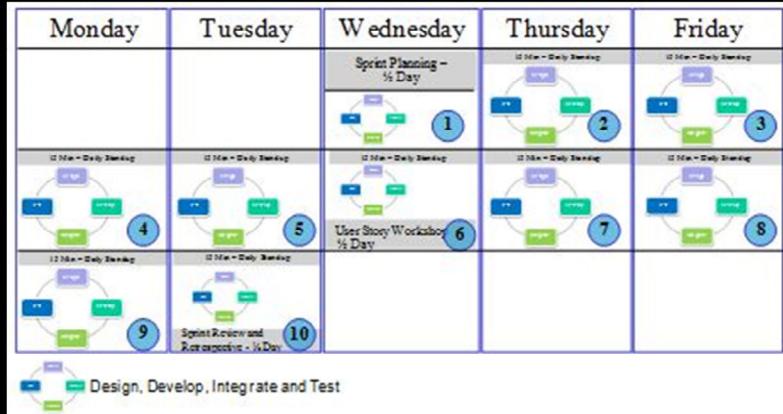
Mon	Tues	Weds	Thurs	Fri
5	6	1 1-Hr Pre-Meeting for Next Sprint	2	3 Sprint Review & Retrospective
6 Sprint Planning Meeting	7 1	8 2	9 3	10 4
13 5	14 6	15 7	16 8	17 Sprint Review & Retrospective
20 Sprint Planning Meeting	21 1	22 2	23 3	24 4
27 5	28 6	29 7	30 8	31 Sprint Review & Retrospective

8/25/2023 8:27 PM

vijaynathani.github.io

Page 20

Cadence

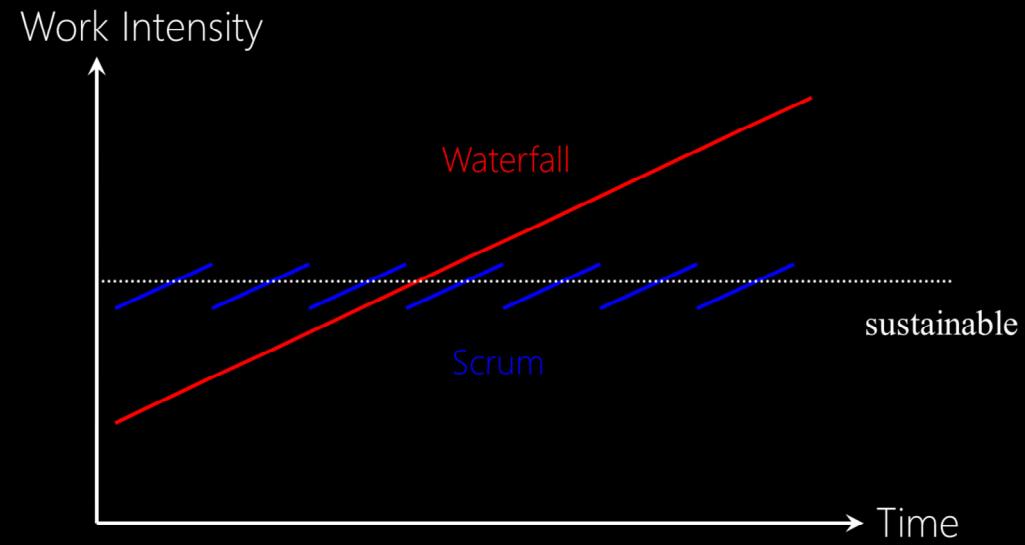


8/25/2023 8:27 PM

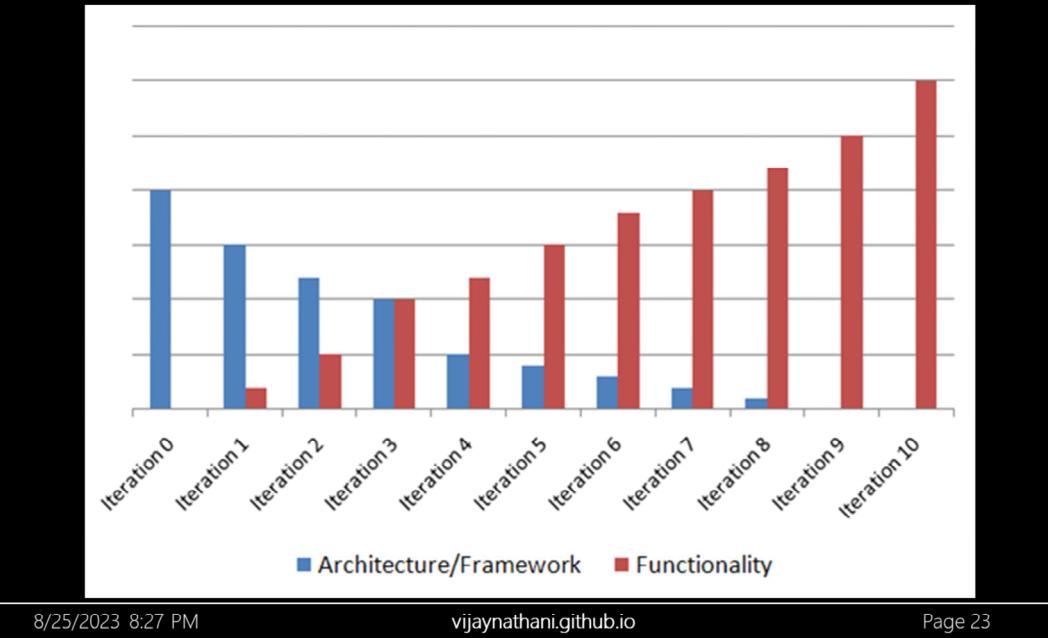
vijaynathani.github.io

Page 21

Work Intensity



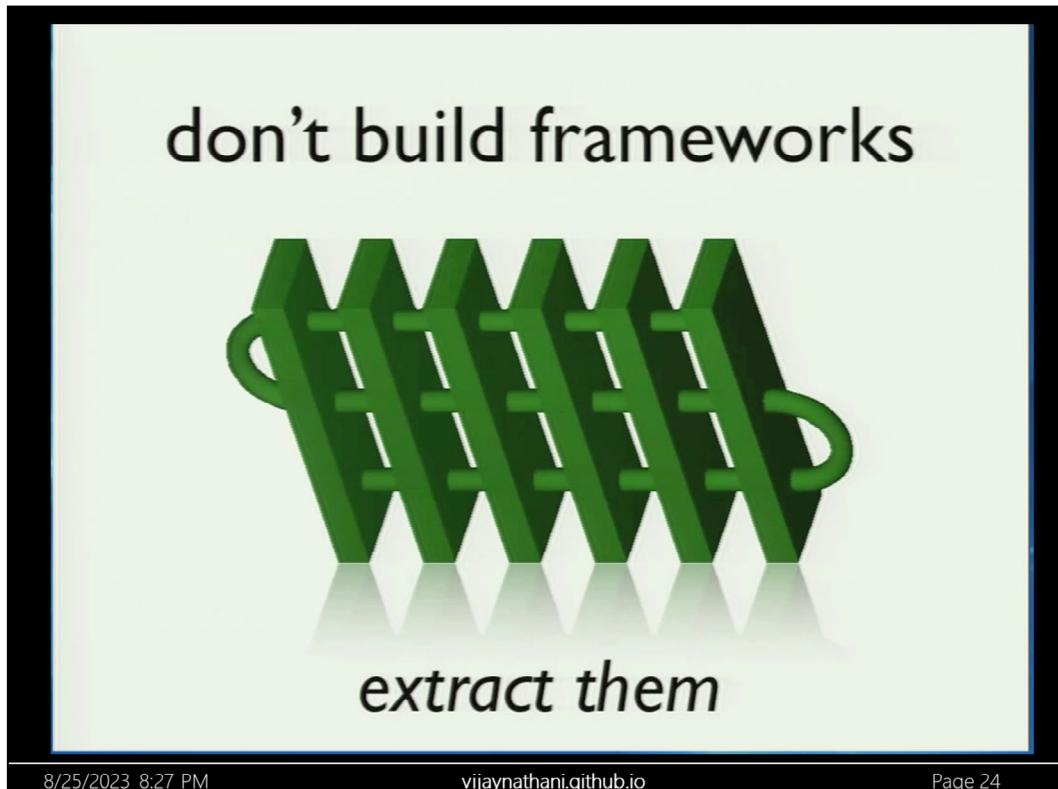
Architecture emerges



8/25/2023 8:27 PM

vijaynathani.github.io

Page 23



Developers love to build frameworks.

In Agile, instead of building frameworks, we extract them from working code.

The best frameworks emerge from working application.

The Basics of Scrum



No Changes to Deliverable

Once team has committed, no changes

Details will emerge during Sprint, but no new work or substantially changed work

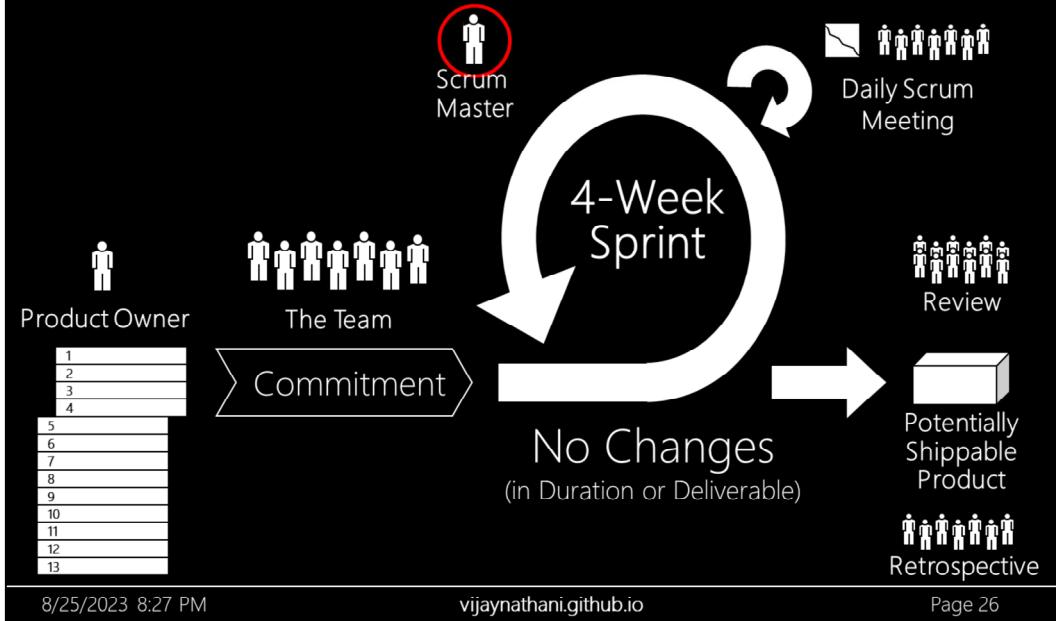
Product Owner can terminate the Sprint if necessary

The experience of the Product Owner?

No Changes to Sprint Duration

Sprint ends on planned date whether team has completed its commitment or not

The Basics of Scrum



ScrumMaster

- Watch the process a whole and
- Calls the team's attention to impeding problems or opportunities for improvement



8/25/2023 8:27 PM

vijaynathani.github.io

Page 27

As "XP coach" defined by Kent Beck in "Extreme Programming Explained"

I'd just finished a speech to my team about how we needed to do something. There were no objections or alternative suggestions and that closed the issue. We could move on to the next topic, and I undoubtedly told the team how they could solve that problem, too.

Fortunately, this was many years ago. And I realized by the end of that meeting that my job as the Scrum Master was not to make decisions for the team. Instead, it was to help *them* make decisions.

And I decided the best way for me to do that would be to stop making so many statements like "We should do such-and-such." And instead, I should be asking more questions.

I began asking questions like:

What's the worst thing that could happen if we made that decision?

What are three other alternatives you considered before choosing this path?

What information would be helpful to have in making a decision?

What's the smallest thing we can do to validate that decision?

And this helped the team. They gradually stopped looking to me to always have an answer or solution. At first that slowed us down. But the team quickly became used to this and started to embrace owning their decisions.

To help me stick to my plan of asking more questions, I started keeping track of how often I made a statement and how many times I asked a question. I didn't track *sentences*. Sometimes a statement or question would take more than one sentence.

ScrumMaster

- Does whatever it takes to make the Scrum Team successful
 - Removing organizational impediments
 - Facilitating meetings
 - Acting as gatekeeper



ScrumMaster

- “The job of Scrum Master is to manage the process. They are not supposed to make product or technology decisions.”
- No authority and not responsible for the success of the project.



8/25/2023 8:27 PM

vijaynathani.github.io

Page 29

A manager manages tasks. ScrumMaster is a leader not a manager.

The **ScrumMaster** is a facilitative team leader working closely with the Product Owner. He must:

- Ensure that the team is fully functional and productive;
- Enable close cooperation across all roles and functions;
- Remove barriers;
- Shield the team from external interferences; and
- Ensure that the process is followed, including issuing invitations to Daily Scrum, Sprint Review and Sprint Planning meetings.

The ScrumMaster has three primary responsibilities in addition to leading the Daily Scrum meeting:

1. The ScrumMaster needs to know what tasks have been completed, what tasks have started, any new tasks that have been discovered, and any estimates that may have changed. This makes it possible to update the Burndown Chart which shows the cumulative work remaining day by day. The ScrumMaster must also look carefully at the number of open tasks in progress. Work in progress needs to be minimized to achieve lean productivity gains.
2. The ScrumMaster needs to surface dependencies and blocks which are impediments to the Scrum. They need to be prioritized and tracked. A remediation plan needs to be implemented for impediments in priority order.

Some can be resolved with the team, some can be resolved across teams, and others will need management involvement as they may be company issues that block all teams from achieving their production capacity. For example, a telecom company recently implemented Scrum and found eighteen items on their impediment list, only two of which were directly related to Scrum teams. The others were company issues that needed management attention.

3. Last but not least, the ScrumMaster may notice personal problems or conflicts within the Scrum that need resolution. These need to be clarified by the ScrumMaster and be resolved by dialogue within the team, or the ScrumMaster may need help from management or the Human Resources. Certified ScrumMaster James Coplien developed over 200 case studies of notable projects while working at ATT Bell Labs. He reports that over 50% of productivity losses were caused by personnel issues. The ScrumMaster must pay attention to them to ensure the team is fully functional and productive.

The Role of the ScrumMaster

- Serve the Team
- Protect the Team
- Support the Team's use of Scrum

Serve the Team

Teaches the Product Owner and the Team

The ScrumMaster facilitates the removal of impediments to the team's productivity

The ScrumMaster facilitates the team's group interactions, to help the team achieve its full potential

ScrumMaster



Bureaucratic Manager says: Follow the rules

Task Manager says: Here is what to do and how – to do it.

Group Facilitator says: You're empowered

Builder of Learning Organization: Here is our purpose and direction. I will guide and coach. (This is the ScrumMaster – a Lean Leader)

ScrumMaster

- Enforces Scrum Practices.
- Coaching
 - Doesn't control the roadmap.
 - Doesn't control the team.
- Removes impediments.
- Not a line manager.
- Not a team leader.
- Not a tech guru.

8/25/2023 8:27 PM

vijaynathani.github.io

Page 32

ScrumMaster maintains an impediment backlog. It contains both technical and personal issues.

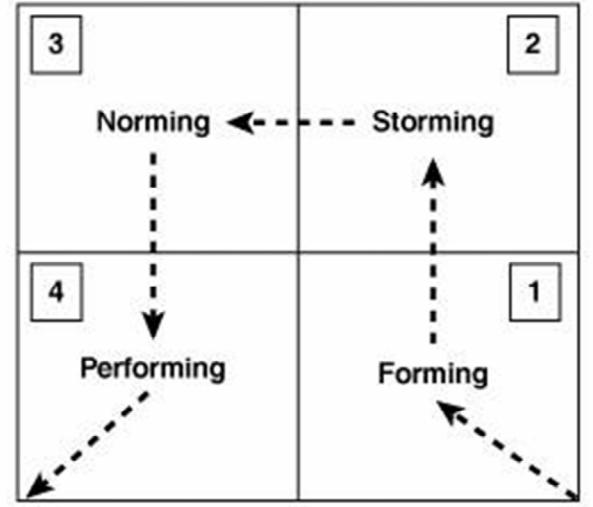
Is ScrumMaster a full time role? Yes, if the company has never done Agile projects or if team members are new to Agile.

When people have done multiple Agile projects successfully, then time taken of ScrumMaster

	Team size < 3
	Team size > 7
Few Problems	10%
	50%
Many Problems	50%
	100%

Phases of Growth

Forming Storming Norming Performing Model



8/25/2023 8:27 PM

vijaynathani.github.io

Page 33

No true team can skip these four phases of growth.

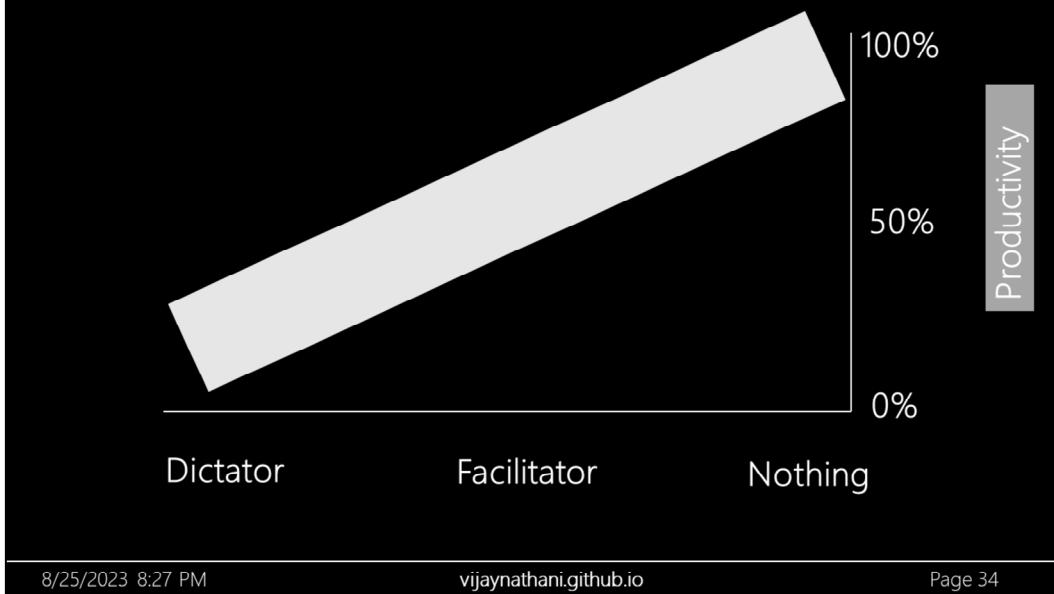
- **Forming** Group members are learning how each of them works and are trying to figure out who will play what role in the team. Such teams need strong facilitative leadership and guidance to keep them on track.
- **Storming** The jockeying for power and control begins, and team members express disagreement, distrust, and prejudice as they form alliances. Divergence is the primary characteristic of a Storming team, and so the group leader must be adept at conflict management in order to shepherd the team through this phase of discomfort.
- **Norming** Team members now have a sense of consensus and the ability to make decisions. They trust one another about how to delegate work to sub-teams. The leader exerts less and less influence and acts mainly as a facilitator. Conflicts are resolved quickly and without destruction.
- **Performing** The team now can focus its full attention on the performance challenge; team culture is healthy and self-managing. The team can self-navigate its work within a larger vision or strategy. All the characteristics of a Katzenbach-Smith high-performing team are in place: trust, sharing, mutual accountability, self-organization, and commitment.

Initially, the team will be in forming stage. Moving towards Storming is ScrumMaster's responsibility.

To achieve consensus, the ScrumMaster guides the team towards Norming.
After being in Norming stage for some time, they will move towards Performing stage. Now the ScrumMaster is mostly free and take other projects.

- Typically at Microsoft they found that it takes 3 months for a new team to go to norming/performing stage. If the team is distributed then it takes about double that time.

Different ScrumMasters



When productivity is much less than 50%, ScrumMaster is a dictator.

When Productivity is near 50%, ScrumMaster is a facilitator.

When productivity is near 100% (15 function points / developer month), then ScrumMaster only watches. He/She intervenes only on a need basis.

Doing nothing is like a captain in the cockpit of huge aircraft. Even though the plane is on auto-pilot, the captain watches the indicators.

In Japanese martial art,

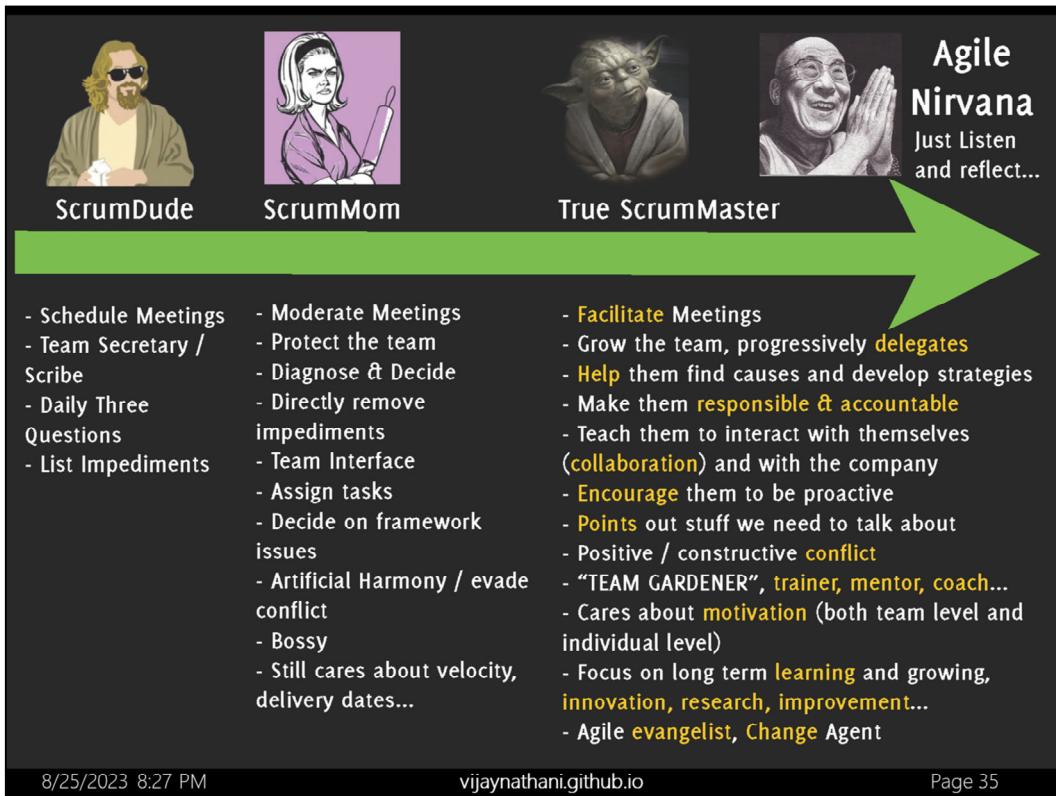
Shuhari roughly translates to "first learn, then detach, and finally transcend."

shu "protect", "obey" — traditional wisdom — learning fundamentals, techniques, [heuristics](#), proverbs. Here ScrumMaster is a dictator.

ha "detach", "digress" — breaking with tradition — detachment from the illusions of self. ScrumMaster is a facilitator.

ri "leave", "separate" — [transcendence](#) — there are no techniques or proverbs, all moves are natural, becoming one with spirit alone without clinging to forms; transcending the physical. ScrumMaster does nothing, but watch.

A good ScrumMaster should be able to move from Dictator to Nothing in maximum 10 sprints. (See <https://www.youtube.com/watch?v=K19m8fXwEEY>)



8/25/2023 8:27 PM

vijaynathani.github.io

Page 35

Guide & Track Team Meetings

- Set time box for meeting
- Set ground rules
- Track decisions that come out of the meeting
- Declare how results will be communicated

How to stay on topic? ScrumMaster holds up a red card.

Have we discussed this issue before?

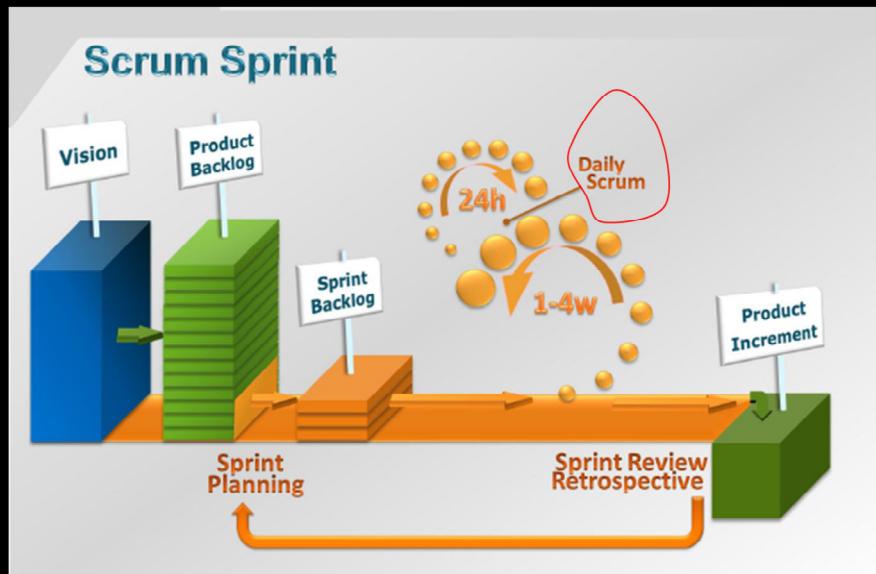
When do Agile teams plan? Every iteration, everyday.

ScrumMaster

- Good Communication
- Calm, not easily panicked
- Technically skilled
- Knows the process; can explain to managers.

“XP Coach” by Kent Beck in “Extreme Programming explained”

Daily Scrum



8/25/2023 8:27 PM

vijaynathani.github.io

Page 38

Daily Scrum Meeting



8/25/2023 8:27 PM

vijaynathani.github.io

Page 39

It is not a place to discuss the solution to the problems.

Any impediments that are raised become the ScrumMaster's responsibility to resolve as quickly as possible. Typical impediments are:

- * My ____ broke and I need a new one today.
- * I still haven't got the software I ordered a month ago.
- * I need help debugging a problem with ____.
- * I'm struggling to learn _____ and would like to pair with someone on it.
- * I can't get the vendor's tech support group to call me back.
- * Our new contractor can't start because no one is here to sign her contract.
- * I can't get the ____ group to give me any time and I need to meet with them.
- * The department VP has asked me to work on something else "for a day or two."

In cases where the ScrumMaster cannot remove these impediments directly himself (e.g., usually the more technical issues) he still takes responsibility for making sure someone on the team does quickly resolve the issue.

Daily Scrum Meeting

- Every weekday, same time and place
- Whole team attends
- Everyone stands
- Lasts 15 minutes or less
- Everyone reports 3 things only to each other
 - "What was I able to accomplish since last meeting"
 - "What will I try to accomplish by next meeting"
 - "What is blocking me"
- No discussion, conversation until meeting ends

8/25/2023 8:27 PM

vijaynathani.github.io

Page 40

Note: We are interested in what you have accomplished and not what you are working on?

ScrumMaster can setup a penalty e.g. Rs 25 for late coming Rs 50 for not coming. The money goes to charity. No direct benefit to the team like party, gifts, etc from the amount collected.

If the meeting finishes early, don't waste time talking about just anything. Maximum 15 minutes. Can finish early.

Only pigs can talk; Chickens are welcome, but they must remain silent.

=====

Not for Problem solving

Following the Daily Scrum Meeting...

Team members update the Sprint Backlog with hours remaining for tasks that they've worked on

ScrumMaster adds up the hours remaining and plots it on the Burndown Chart

Smells for Daily Scrum Meeting

- Reporting to the Leader and not the Team
- People are late / Low energy
- Stand-up meeting starts the day
- Observers interrupt
- Socializing
- I can't remember
- Story Telling
- Problem solving
- Obstacles not raised / not removed / only raised here

With Scrum

- Scrum is based on TRUTH, TRANSPERANCY and COMMITMENT

Know where you are every day with Scrum

- or -

Think you know where you are on your well-formed plan and discover that you are very wrong, very much later

In a world of universal deceit, telling the truth is revolutionary. – George Orwell, 1984.

8/25/2023 8:27 PM

vijaynathani.github.io

Page 42

WITHOUT SCRUM:

Management actually believes that it can predict the cost, delivery schedule, and functionality that will be delivered, and plans accordingly.

Developers and project managers are forced to live a lie: they pretend they can plan, predict and deliver.

They build one way, but must pretend they build another way. In the end, they are essentially without controls.

By the time the system is delivered, it is often irrelevant or requires significant change.

A key cause is that high iteration costs limit our visibility into the usefulness of what the team is actually developing, until it is too late.

Sprint Backlog

Task	Task Owner	Hours of Work Remaining on Each Day of the Sprint									
		Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
Configure database and space IDs for Trac	Sanjay	4	3	3	1	0					
Use test data to tune the learning and action model	Jing	2	1	1	1	1					
Setup a cart server code to run as apache server	Philip	2	2	5	2	0					
Implement pre-Login Handler	Tracy	6	3	2	2	2					
Merge DCP code and complete layer-level tests	Jing	8	5	4	2	2					
Complete machine order for pRank	Jing	4	4	3	2	3					
Change DCP and reader to use pRank http API	Tracy	2	2	0	0	0					
	Total	50	48	44	43	34					

8/25/2023 8:27 PM

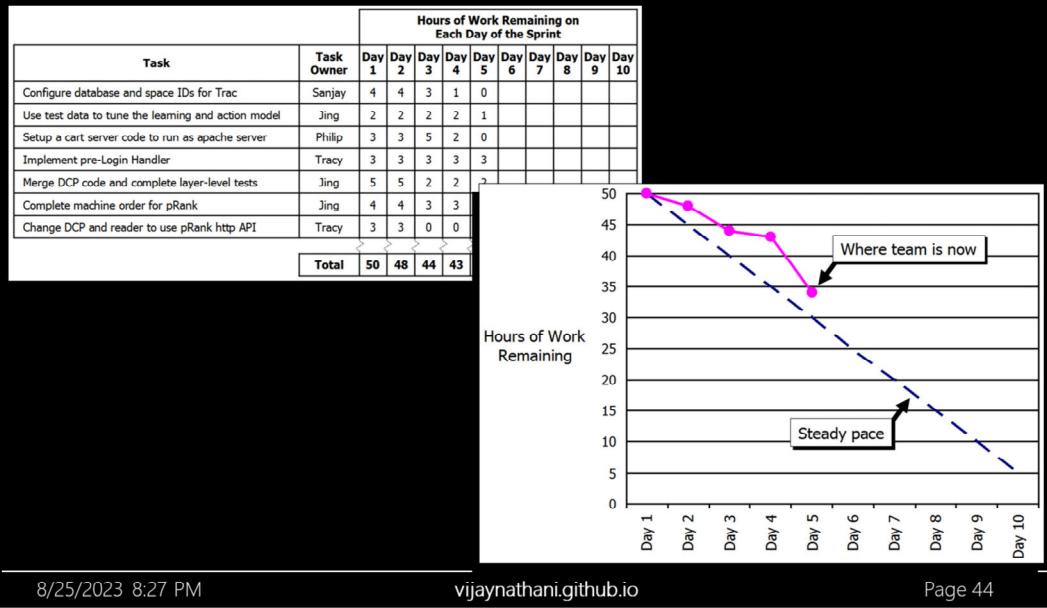
vijaynathani.github.io

Page 43

This is usually updated before / after the daily scrum meeting.

Stop the line culture. If a team thinks that it is not going to meet the Sprint backlog, abort the Sprint.

Sprint Burndown Chart



8/25/2023 8:27 PM

vijaynathani.github.io

Page 44

Sprint Backlog

Changes

Team adds new tasks whenever they need to in order to meet the Sprint Goal

Team can remove unnecessary tasks

But: Sprint Backlog can only be updated by the team

Estimates are updated whenever there's new information

Task Board

TO DO	IN PROGRESS	DONE
<p>Task: Configure database and SpaceIDs for Trac</p> <p>Owner: Sanjay</p>		

8/25/2023 8:27 PM vijaynathani.github.io Page 45

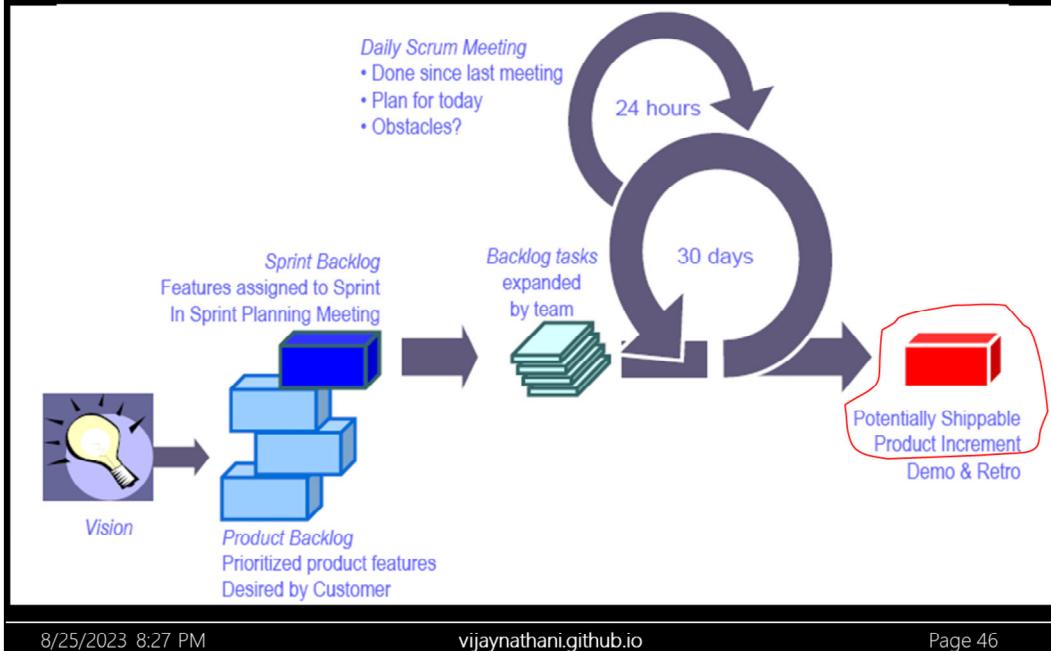
Paper vs. Electronic

Electronic may look more convenient for the Sprint Backlog and Burndown Charts, but...

Paper is fast and easy

Paper is much more visible

Scrum



8/25/2023 8:27 PM

vijaynathani.github.io

Page 46

Potentially Shippable Product

- The goal of the Sprint is to produce Potentially Shippable Product

8/25/2023 8:27 PM

vijaynathani.github.io

Page 47

This is the outcome of the Sprint that is of the greatest possible value to the Product Owner or customer.

Anything produced that is less than “potentially shippable” represents unwanted risk for the Product Owner: “Inventory of product that is always not-quite-done”.

However, producing Potentially Shippable Product every Sprint is hard

For many teams, it will require a change in underlying software practices.
For example: putting in place automated testing.

These practices will produce further benefits of productivity and quality for almost all teams

=====

No such thing as 95% done. The shippable product is either 100% or 0% done at the end of sprint.

It includes

- Code
- Tests
- Documentation
- Installation script
- Etc.

=====

Can we use Scrum and not produce Potentially Shippable Product?

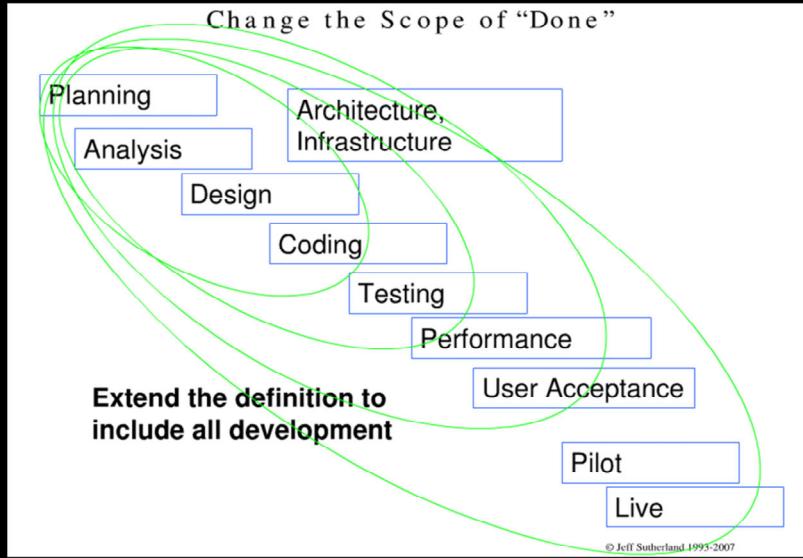
Yes, but you'll be missing a big part of the potential benefit. Keep asking:

"Would it be of value to the customer and our business if we were able to produce Potentially Shippable Product every Sprint?"

"What would we have to do differently if we wanted to be able to produce Potentially Shippable Product every Sprint?"

"Would those changes also in and of themselves benefit the customer or our business?"

Definition of Done



8/25/2023 8:27 PM

vijaynathani.github.io

Page 48

Initially the team may be unable to deliver production grade software. So they may deliver unit tested software.

As iterations proceed, they will have to refine the definition of done.

The above slide has been taken from a presentation from Infoq by Jeff Sutherland.

Delivery vs Deployment

Continuous Delivery: **Can** deploy to Production

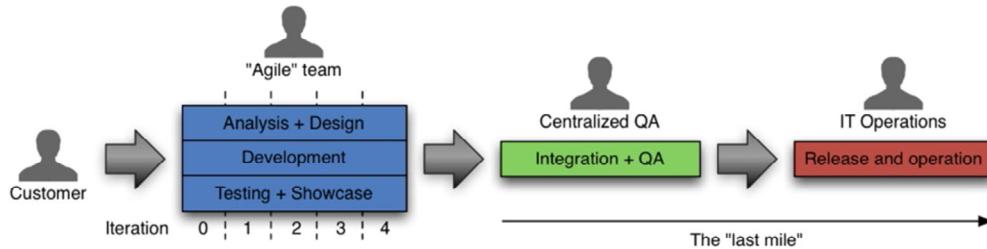


Continuous Deployment: **Do** deploy to Production

Deployment is a business decision.

The goal of Agile is “Continuous Delivery”. After this goal is achieved, within a short **FIXED** time duration, deployment should be possible anytime.

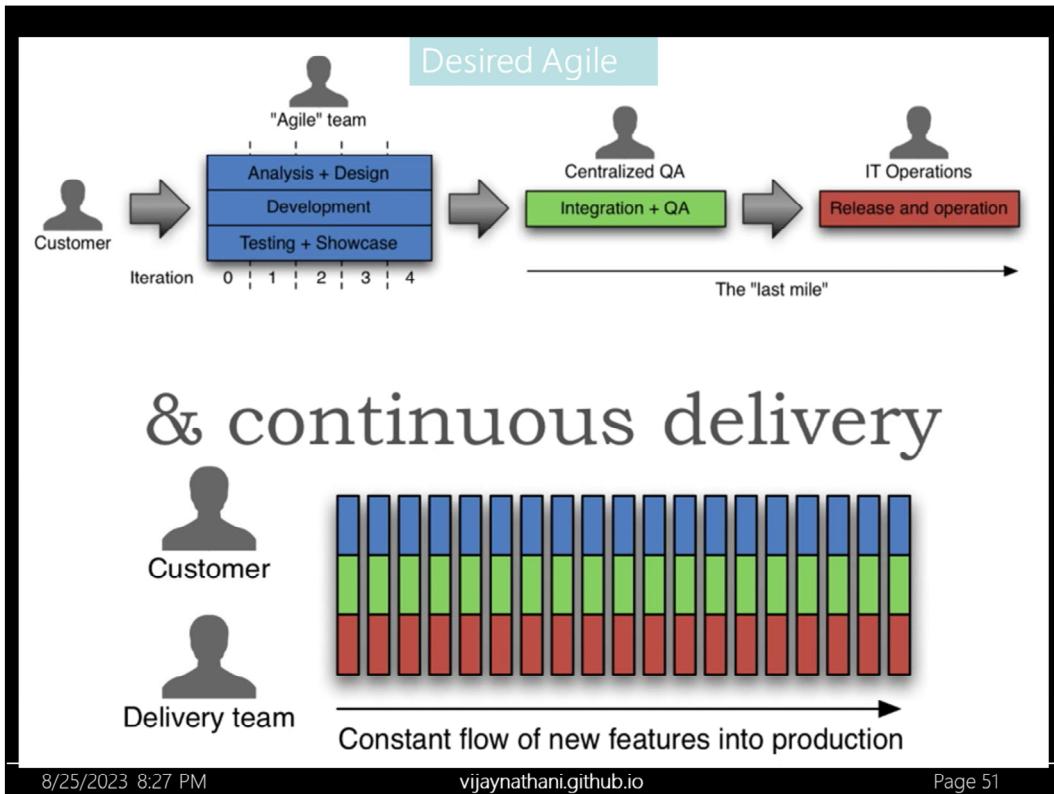
Basic Agile



8/25/2023 8:27 PM

vijaynathani.github.io

Page 50



Amazon Deployment Stats for typical month

11.6 seconds

Mean time between deployments (weekday)

1,079

Max # of deployments in a single hour

10,000

Mean # of hosts simultaneously receiving a deployment

30,000

Max # of hosts simultaneously receiving a deployment

Jon Jenkins, "Velocity Culture, The Unmet Challenge in Ops" | <http://bit.ly/1vjo1Ya>

8/25/2023 8:27 PM

vijaynathani.github.io

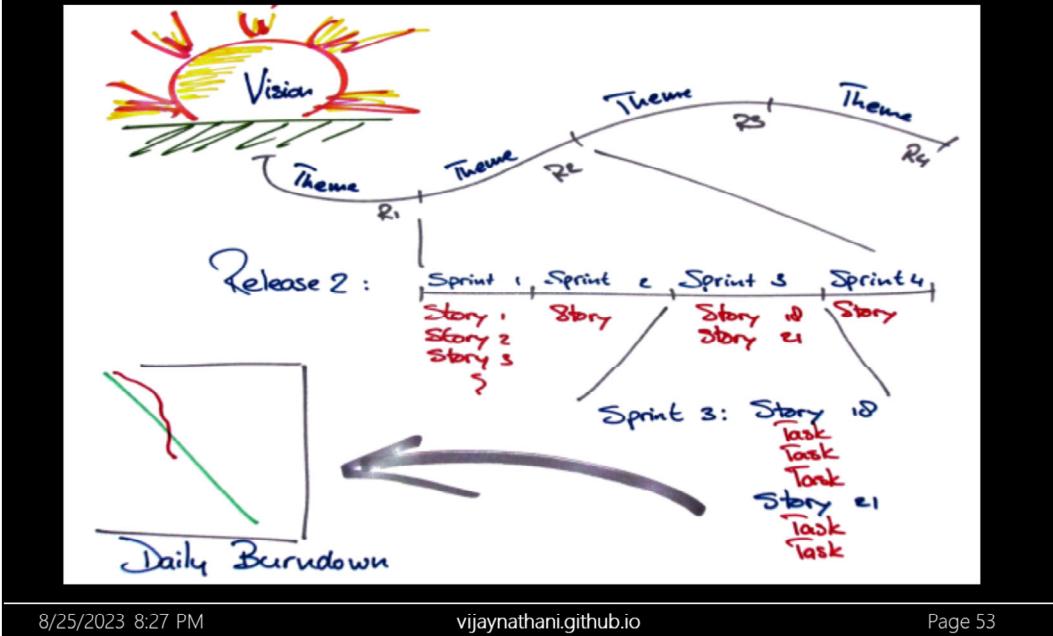
Page 52

The above stats are for one typical month considering production hosts/environments only.

Rapid feedback means do less.

"Evaluating well-designed and executed experiments that were designed to improve a key metric, only about 1/3 were successful at improving the key metric." – "Online experimentation at Microsoft", Kohavi,
<http://standford.io/130uW6X>. Kohavi was engineer at Amazon and then he shifted to Microsoft.

Add Detail where it matters



8/25/2023 8:27 PM

vijaynathani.github.io

Page 53

A group of User stories form a Theme.

A group of Themes form an Epic.

The Basics of Scrum



Sprint Review

- The Sprint Review takes place at the end of each Sprint
- Purpose of the Sprint Review is
 - Demo what the team has built
 - Generate feedback, which the Product Owner can incorporate in the Product Backlog
- Attended by Team, Product Owner, ScrumMaster, functional managers, and any other stakeholders
- A demo of what's been built, not a presentation about what's been built
 - No PowerPoint allowed!
- Usually lasts 1-2 hours
- Followed by Sprint Retrospective

8/25/2023 8:27 PM

vijaynathani.github.io

Page 55

It's called the "sprint review meeting" for a reason. During this meeting, stakeholders, the product owner and the rest of the team *review* what was completed during the sprint.

It's not *just* a demo, so please don't call it the "sprint demo." Yes, demonstrating the work that was completed during the sprint is the most visible part of a sprint review. But it's not just a demo.

I think of a demo as one-directional. I think of a demo as the salesperson at a tradeshow carefully showing what a product will do by following a scripted path through the product's functionality.

Whereas as a demo is one-directional, a review is two-directional. Instead of just showing the product, a review includes discussion of what meeting participants think of the product.

Calling the "Sprint Review" as "Showcase" is bad because it implies the review should be fancier or more formal than it really should be. I need to dress up for the showcase. Maybe put on a tie. Get my shoes shined. At a review, some team members may not even wear shoes!

The sprint review should not be a formal, fancy meeting. It's a point-in-time snapshot of where the product is as of the last day of the sprint.

When a team calls it the showcase, they tend to think of the review as being more than this. The same is true when it's called "the dog and pony show" as I've encountered a few times.

But I've heard the sprint review called worse. I worked with a team that called it the "Christian and the lions meeting." The team felt like they were being chewed apart by their stakeholders every two weeks and used that name informally. Renaming the sprint review changes how everyone thinks of the meeting even if only in small, subtle ways. It's possible there's a better name for this meeting, but I haven't heard it yet. And it's definitely not one of those I've listed here.

By calling it the sprint review, you'll help your team understand that the meeting is not just a demo, showcase or dog and pony show.

What is Wrong here?

- The Sprint Review goes well. Senior management is there to encourage the team. At the end of the Sprint Review, the CEO and everyone present applaud the team for their fine work.

PO decides Sprint success



The Basics of Scrum



Agile Law

- Failure to learn is the only true failure



8/25/2023 8:28 PM

vijaynathani.github.io

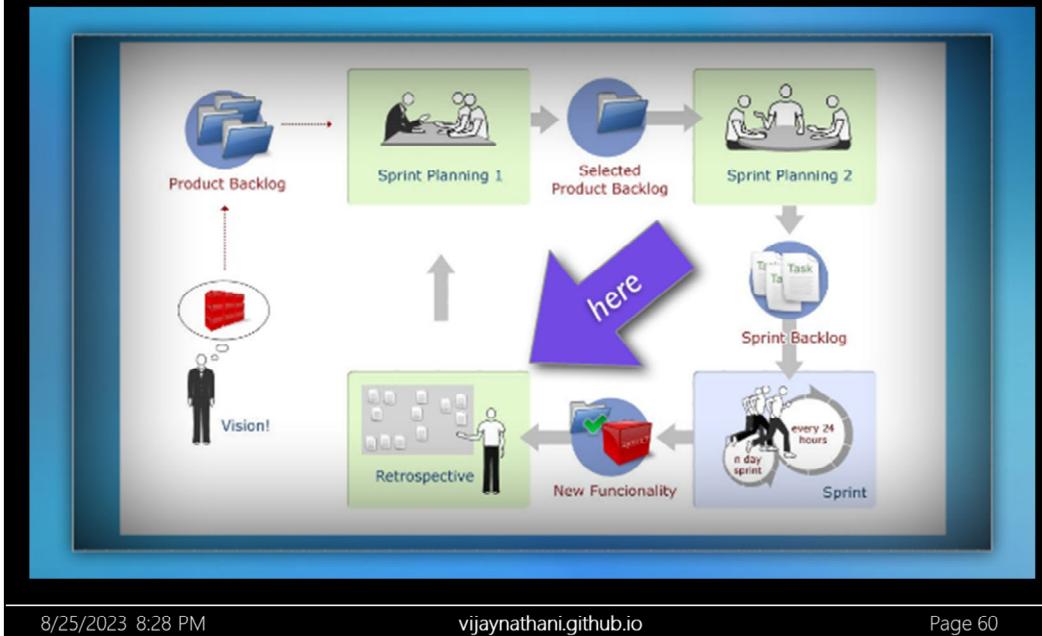
Page 59

Demming cycle: Plan, Do, Check, Act

Learning from Past for the future.

Disappointments (of expectations) stops learning.

When is the best time?



8/25/2023 8:28 PM

vijaynathani.github.io

Page 60

Retrospective should be done in a dedicated room. No interruptions.

Sprint Retrospective

- What is it?
 - 1-2 hour meeting following each Sprint Demo
 - Attended by Product Owner, Team, ScrumMaster
 - Usually a neutral person will be invited in to facilitate
 - What's working and what could work better

8/25/2023 8:28 PM

vijaynathani.github.io

Page 61

Time boxed to 3 hours.

A rough guide to timings is a team need 30 minutes retrospective time per week under review so using this formula allow 2 hours for a monthly retrospective and a whole day for a retrospective of a several months work.

Team and ScrumMaster talk about the last Sprint.

This meeting may not be attended by Product Owner, if the team feels it will be counter-productive.

Any other person can be invited if the Team so wishes.

Scrum Smells

- Loss of Rhythm
- Persistent Signatures
- ScrumMaster assigns work
- The daily Scrum is for ScrumMaster
- Specialized Job Roles

Loss of Rhythm symptom: Sprints are not always of same length.

Persistent Signatures symptom: The wild fluctuations shown on a team's initial sprint burndown charts continue to be seen in much later sprints.

What if Retrospective doesn't work?

- Ensure that everyone feels safe
 - Take a secret vote.
- Mention that Knowledge of the project is not being measured, its your comfort in talking in the room.

8/25/2023 8:28 PM

vijaynathani.github.io

Page 63

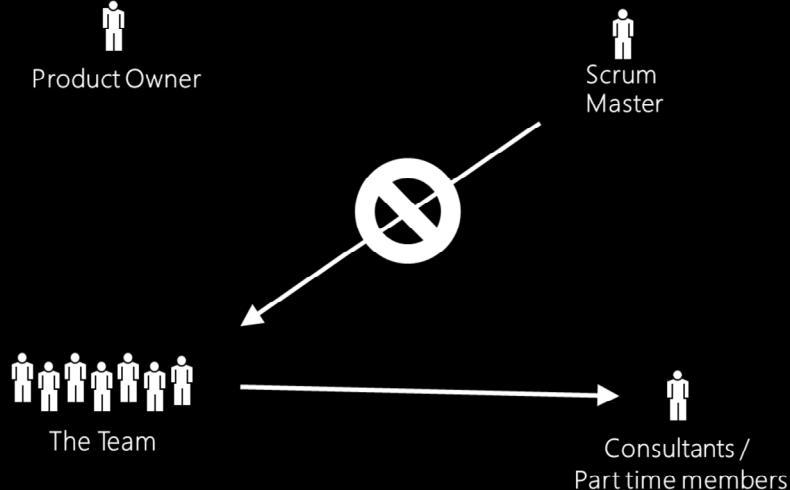
For an honest discussion, everyone must feel safe.

Use a Secret vote, where every team member writes 1 to 5 on a post-it note

- 5) No problem. I will talk about anything
- 4) I will talk about almost anything
- 3) I will talk about some things, but others will be hard.
- 2) I won't say much. I will let others bring up the issues.
- 1) I will smile, claim that its great and agree with seniors around.

Ensure that almost everyone is 3+.

Evaluation?



8/25/2023 8:28 PM

vijaynathani.github.io

Page 64

At cisco, the business person decides the features and evaluates the Team.

At ScrumWorks, the team is evaluated as a unit.

At Google, there is no evaluation of anyone. Each employee is supposed to maintain a web page which says

- What am I doing now?
- What are my objectives in the next months?

Menlo Innovations: Company takes a % of total revenue. The rest is decided by team regarding how to distribute among themselves.

Module: Software Development Practices in Scrum

Theory, by itself, teaches nothing.

Application, by itself, teaches nothing.

Learning is a dynamic interplay between the two.

- Peter Scholtes in “The Leader’s handbook”

Engineering Practices

- Use Scrum with XP for software development

Module: Using Scrum in Contract Software Development

A new doctrine goes through three stages. It is attacked and declared absurd; then it is admitted as true and obvious but insignificant. Finally, its true importance is recognized and its adversaries claim the honor of having discovered it. — William James

Contract Terms

the party in the first part shall remunerate the party in

3.1.1 Vendor will demo potentially shippable software to Customer every 30 days. There will be no additional charge for this.

3.1.2 Customer can replace any requirements that Vendor hasn't yet started working on with one or more of equal total size (in the estimate of Vendor) at any time. There will be no additional charge for this.

3.1.3 Customer may request interim releases at any time, and will be charged an agreed-upon time and materials cost.

3.1.4 If Customer's business goals are satisfied early, Customer may terminate contract early for 20% of the remaining unbilled contracted amount.

notwithstanding the foregoing clause, the party in the first

Different Contracts:

- 1) T&M
- 2) Rolling Contract: Renew every 3 months.
- 3) Fixed price, fixed duration but variable features.
- 4) Price, Duration and Features fixed. Then use the above clauses.

Module: Managing Scrum Adoption in the Enterprise

You don't transition your projects to agile projects; you have to transform your people to think in agile ways.

Stages

1. Scrum adopted for a team.
2. Automation within the team.
3. Organizational changes.

In Step 1, many parts are manual

In Step 2, usually TDD and CI are setup.

In Step 3, support is needed from the organization to improve productivity.

Google Strategy

- Adopted company wide Agile process in 2001

When Rosing started at Google in 2001, "we had management in engineering. And the structure was tending to tell people, No, you can't do that." So Google got rid of the managers. Now most engineers work in teams of three, with project leadership rotating among team members. If something isn't right, even if it's in a product that has already gone public, teams fix it without asking anyone.

"For a while," Rosing says, "I had 160 direct reports. No managers. It worked because the teams knew what they had to do. That set a cultural bit in people's heads: You are the boss. Don't wait to take the hill. Don't wait to be managed."

And if you fail, fine. On to the next idea. "There's faith here in the ability of smart, well-motivated people to do the right thing," Rosing says. "Anything that gets in the way of that is evil.

Scrum is Hard

- Scrum surfaces all sorts of nasty stuff
 - Make sure people are prepared
 - Make sure they understand that this is Scrum working, not failing
- Help teams learn from each other
- Be ready to stage a rescue mission
 - There are some problems teams can't solve by themselves

8/25/2023 8:28 PM

vijaynathani.github.io

Page 72

Problems are normal. Don't panic. Don't despair.

Visible problem = Killable problem. Opportunity for improvement

Prioritize and fix problems. One by one.

Look back once in a while and pat yourself on the back.

Good and Bad News

- Culture eats Strategy for Breakfast

The Good News –very few of your competitors will be able to compete with you after you change, and very few of them have the guts to change.

The Bad News –very few enterprises succeed, even with compelling reason. Think of Toyota's warning and GM, Ford, and Chrysler's response.

Remember –Culture eats strategy for breakfast.

Find an Executive Champion

- Hold management accountable at board level to remove impediments.



8/25/2023 8:28 PM

vijaynathani.github.io

Page 74

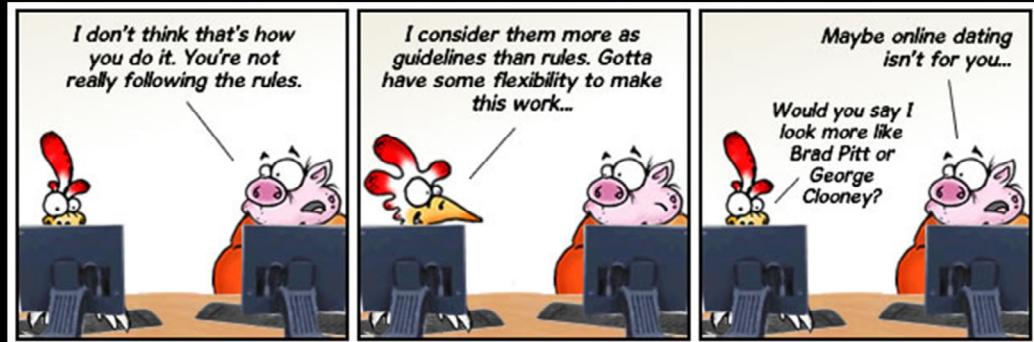
The executive sponsor or champion should clear the path for the pilot teams.

Many champions may feel that once the team has been kicked off their work is done, but this is not so in an agile environment. These executives have to be available to facilitate and drive ongoing improvements in the organization.

There should be an open audience with the executive to inform them of problems that the agile team uncovers. As this is all about continuous improvement, the communication between the teams and the executives is also constant and never-ending.

The Urge to Tinker is Great

- If it isn't Scrum, don't let people call it Scrum
 - Everyone has a way to improve Scrum
 - Protect Scrum's good name!



8/25/2023 8:28 PM

vijaynathani.github.io

Page 75

If it is Scrum, it should have

- Automated testing for new code written / existing code modified. Code coverage for them should be between 80 to 95%.
- Continuous Integration and Tests running multiple times a day.
- Team is self-organizing. No command&control management.
- Production grade software is delivered to the customer regularly at short intervals e.g. every fortnight.
- The team constantly thinks about how to improve itself e.g. Retrospective meeting.

Break any of the above rules and it is not Scrum.

Scrum will Always Be Messy

- Scrum is about people, and people are messy
 - Inconsistent, insensitive, erratic, make mistakes
- For this reason, it will never be perfect
 - Idealists will always be disappointed
 - Opponents will always have ammo
 - Just keep asking, is it better than before

8/25/2023 8:28 PM

vijaynathani.github.io

Page 76

Common Sense:

Problem: Two weeks is too short

Solution: Use one week iteration. Once you have mastered that, two weeks will be a breeze.

Problem: There is not enough room to write requirements on cards.

Solution: Use smaller cards.

Basic Bootstrapping process

- Identify PO and SM
- Channel all requirements through PO
- Create a simple Product backlog
- Identify Scrum Team & seat them together
- Decide on Sprint length and definition of Done.
- Start sprint

Focus and efficiency yields much higher productivity than many heads and long working hours.

Don't worry. Don't wait. Just go.

Don't worry too hard. Just Experiment.

Listen to the team.

Scrum will uncover problems. Don't shoot the messenger.

Prioritize the problems. Fix them one by one.

Last but not the least

- Jump right in, the Water is fine



8/25/2023 8:28 PM

vijaynathani.github.io

Page 78

Don't try to fix everything before you start agile adoption. Just dive right in—you'll find a way to work within the current constraints.

The road to innovation is unpaved

Use the iteration reviews and retrospectives to make change recommendations and implement incremental improvements as you go.

Be reasonable about this by focusing on only the top two or three things for the next iteration.

It's a backlog of recommendations and, like the backlog of product features, they all can't be implemented at once.

It is like in the movie matrix, Morpheus says “Unfortunately no one can be told what the matrix is. You have to see it for yourself.”

Take this Home



"Scrum is like your mother-in-law. It's constantly pointing out your shortcomings." - Ken Schwaber.

The trick is that we're supposed to learn from that feedback and fix our problems.

8/25/2023 8:28 PM

vijaynathani.github.io

Page 79

Make a choice:

Use Scrum OR

Lose business/employees to competition.

Scrum is an effective technique for improving the software development efforts of many professionals. That's it, nothing more.

If you work hard; if you stay focused; if you take Scrum to heart; then you will likely improve your effectiveness as a developer

Agile is facing two issues:

1. Some people are looking for a silver bullet that makes software development easy. They hype Agile beyond its true value to support their hopes and dreams.
2. There is gold in the word Agile. You can make money by touting a development process that improves productivity and lessens the pain of development. If the true definition does not fit, create one that will attract customers. I believe some consultants have done this.

Agile is more like a raft on the raging currents. Work is still required to negotiate the hazards, but the raft was built with the hazards in mind.