



Scrum is not a methodology.

Scrum does not provide answers to building software faster

Scrum is a framework within which the game of product development is played.

The team plays. Scrum makes everything visible.

The team gets to continuously improve itself.

Introduction to Scrum

*Scrum doesn't engender excellence.
It exposes incompetence*

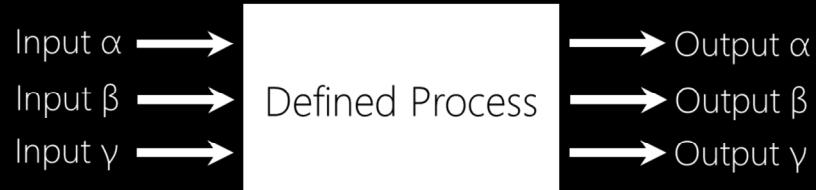
"The problem we face has nothing to do with process and technology, but with people.

Scrum and Agile are based on the hypothesis that there is no meta-solution for software development. Just a framework within which we will be empirical – inspect and adapt.

This is very frustrating to those looking for procedures and final answers."

- Ken Schwaber

Defined Processes



Assuming every step is understood, a given
well-defined set of inputs produces the
same set of outputs every time.
Predictability is key.

Every task must be completely and unambiguously understood

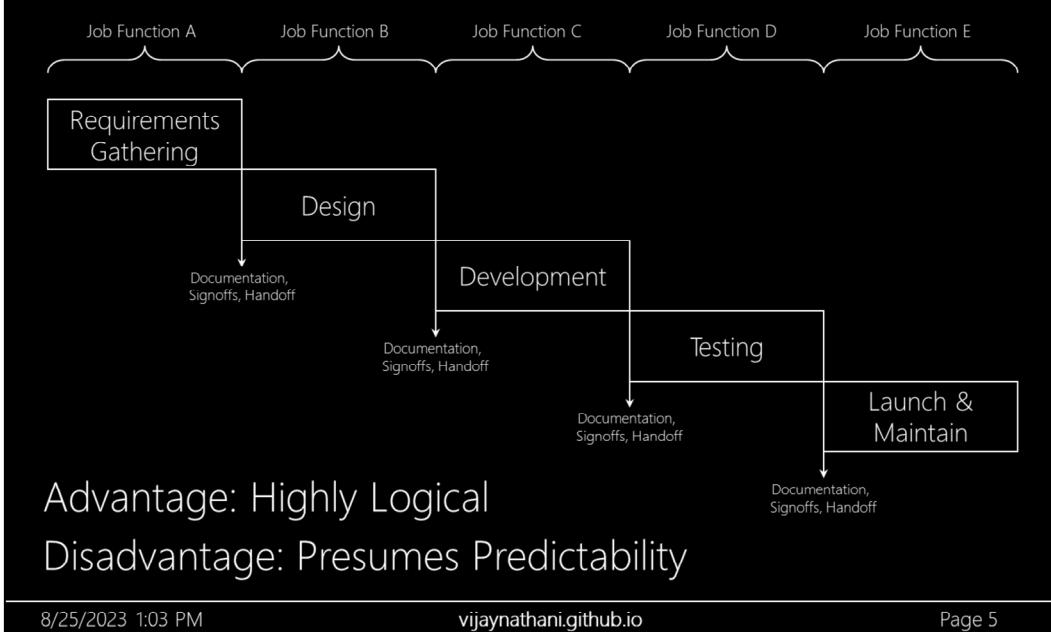
Inputs are completely and unambiguously defined

When given a well-defined set of inputs, the same outputs are generated every time.

Simple processes fall under this category

E.g. Toaster, Opening a lock with a key, clothes dryer, etc

The Waterfall



Defined

- Predictable
- Repeatable
- Known in detail

Is Software Development More Like Ballet or Hockey?



defined



agile

Defined vs. Agile



8/25/2023 1:03 PM

vijaynathani.github.io

Page 7

One is Agile, other is not

Exploration Drilling
(Adaptation)



Production Drilling
(Optimization)





Predictive vs. Agile

Fruit Cake vs Shower knob

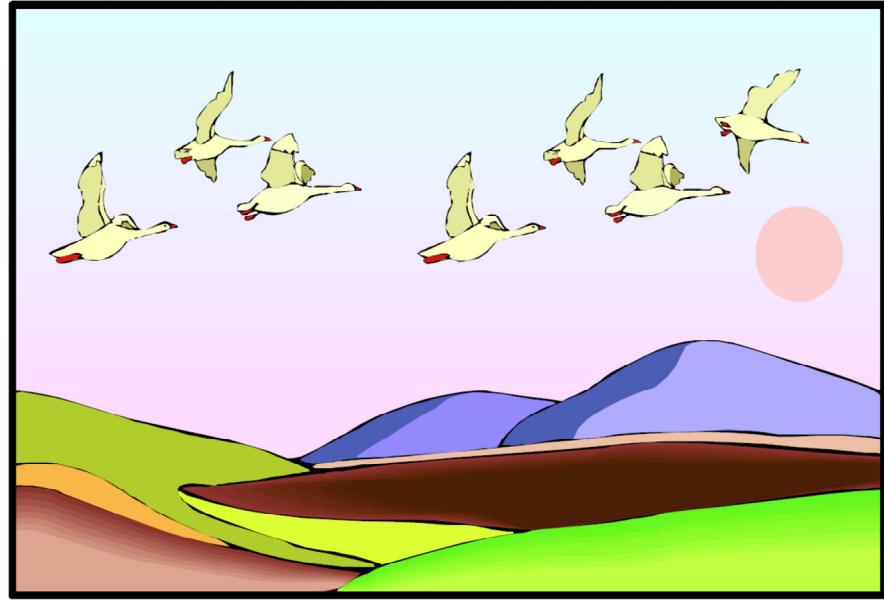


8/25/2023 1:03 PM

vijaynathani.github.io

Page 10

Empirical Process



8/25/2023 1:03 PM

vijaynathani.github.io

Page 11

Geese moving from Canada to some warm place e.g. Florida during winter.
Maybe they will stop at California or Texas.

Other examples of Empirical process:

- Training,
- Having a party
- Ant colony or Bee hive
- A crowd going on a picnic or sporting event.
- A family preparing, eating and cleaning up.

Scrum is a Disciplined Management Methodology

- Wrapper for existing engineering practices
- Maximizes ROI, but doesn't change I
- Scrum works with XP and any other engineering practice you care to entertain



8/25/2023 1:03 PM

vijaynathani.github.io

Page 12

The second point is not obvious if you weren't at the presentation.

"ROI = (Gain – Cost) / Cost" in a given period of time

The complete thought is that while it maximizes your return on investment in software development, it doesn't actually change your investment. In other words, it can't help you if your goal is to hire unskilled programmers and hope that a few so-called "architects" plus a methodology will somehow constrain your team into delivering something.

Scrum maximizes the value from your investment, but it's still important to invest.

Common Examples of Uncertainty and Change

- Uncertainty
 - I won't know if it's right until I see it
 - I don't know what will go wrong
 - We don't know what the competition will do
 - We don't know if the customers will like it
- Change
 - I just had a great new idea
 - The customer just changed his mind
 - The competition just changed his mind
 - Our CEO just changed her mind

8/25/2023 1:03 PM

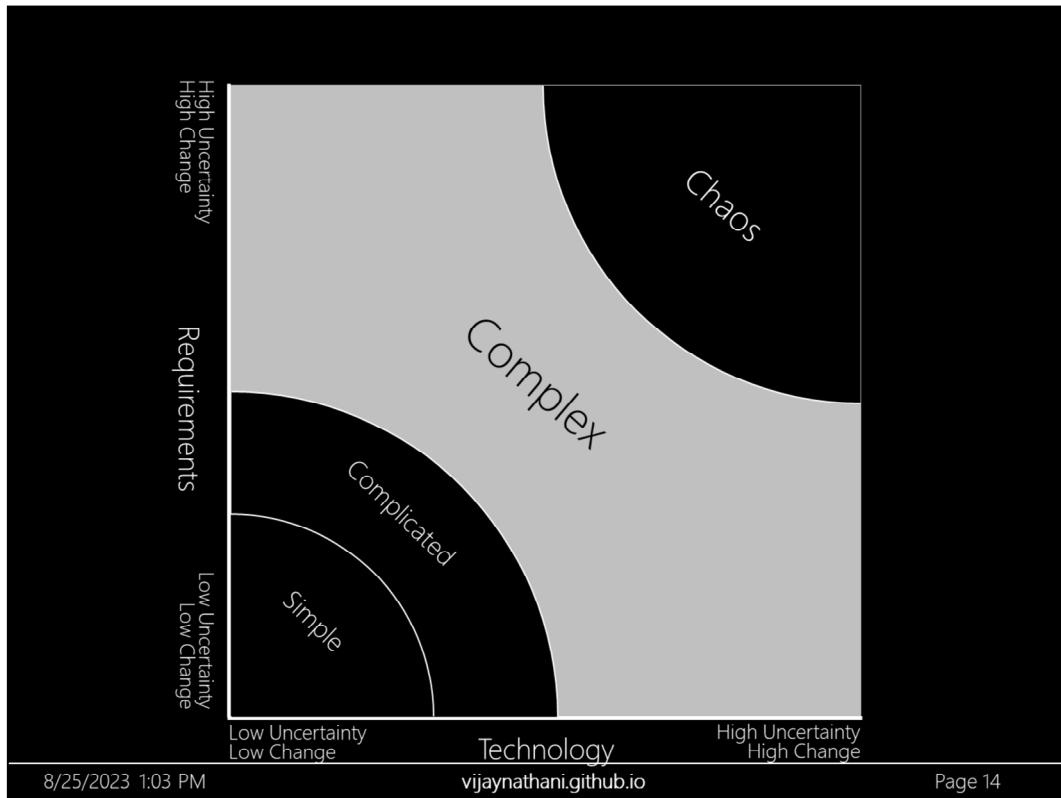
vijaynathani.github.io

Page 13

Empirical

- Unpredictable
- Unrepeatable
- Not known in detail

Examples: Cooking, Driving, Gardening.



People can be on Z axis. Even people vary in their skills and behavior.

For simple projects – defined method works

For complicated/complex projects – empirical process is recommended.

Empirical Processes

- “It is typical to adopt the defined (theoretical) modeling approach when the underlying mechanisms by which a process operates are reasonably well understood. When the process is too complicated for the defined approach, the empirical approach is the appropriate choice.”

Process Dynamics, Modeling, and Control,
Ogunnaike and Ray, Oxford University Press, 1992

- Translation into English: Inspect and Adapt



Empirical Process Control

- Visibility: those aspects of the process that affect the outcome must be visible to those controlling the process.
- Inspection: those aspects of the process that affect the outcome must be inspected frequently enough that unacceptable variances in the process can be detected.
- Adaptation: If the inspector determines from the inspection that one or more aspects of the process are outside acceptable limits and that the resulting product will be unacceptable, the inspector must adjust the process or the material being processed.

8/25/2023 1:03 PM

vijaynathani.github.io

Page 16

Empirical Process:

Effective software development is best implemented via an empirical rather than planned process;

Once organizational impediments are removed, a self organizing and self managing team will naturally deliver better software than would otherwise be the case

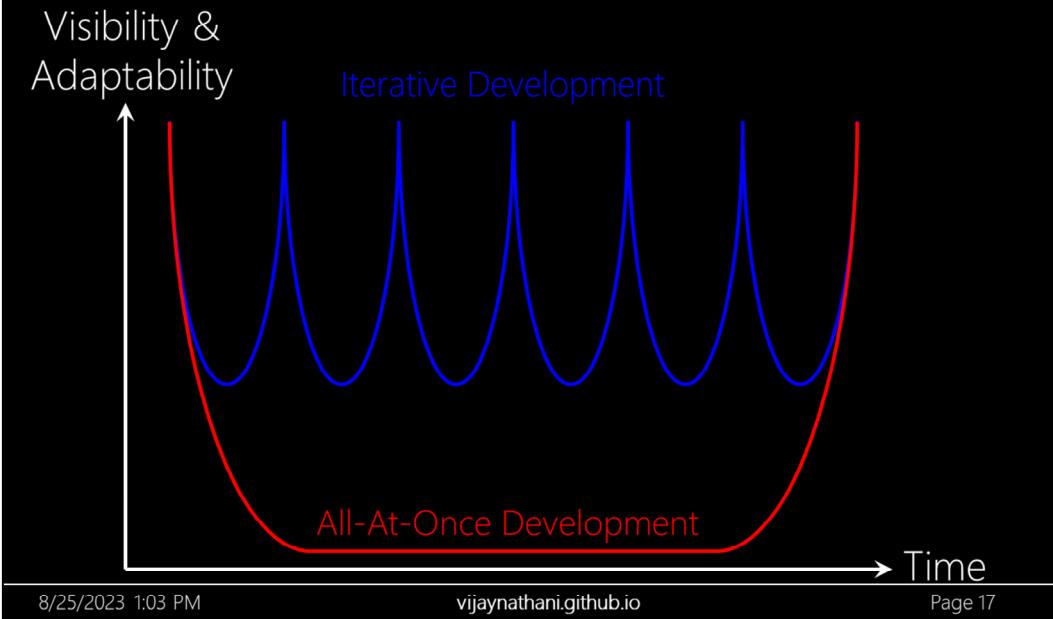
The premise that you can deliver the most valuable software within a prescribed time and budget, and yet you cannot definitively predict the exact functionality of what a team will deliver.

In Predictive processes, developers and project managers are forced to live a lie—they have to pretend that they can plan, predict, and deliver.

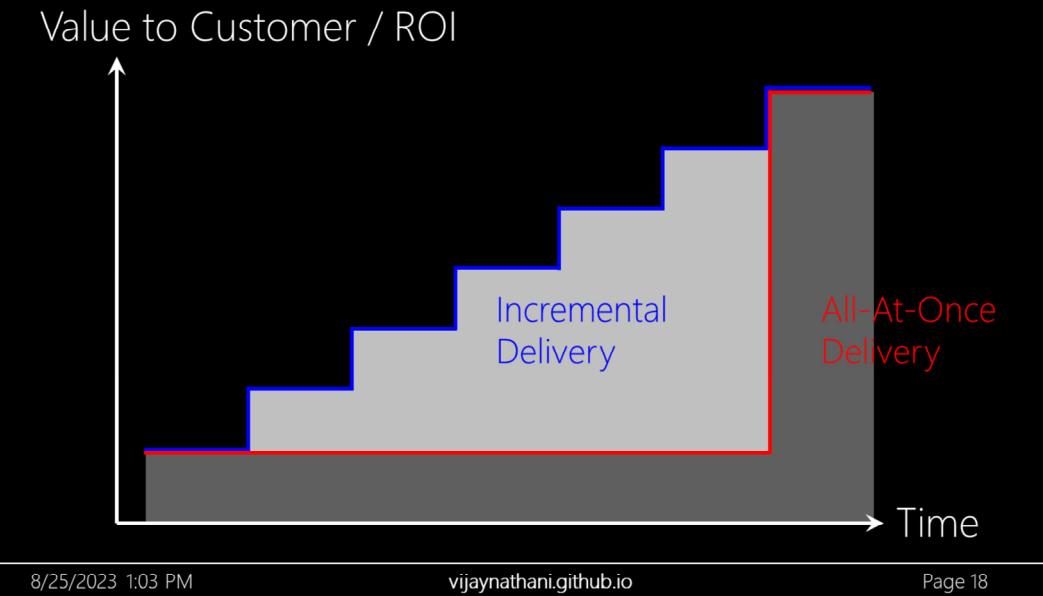
Requirements change. Adapt or die.

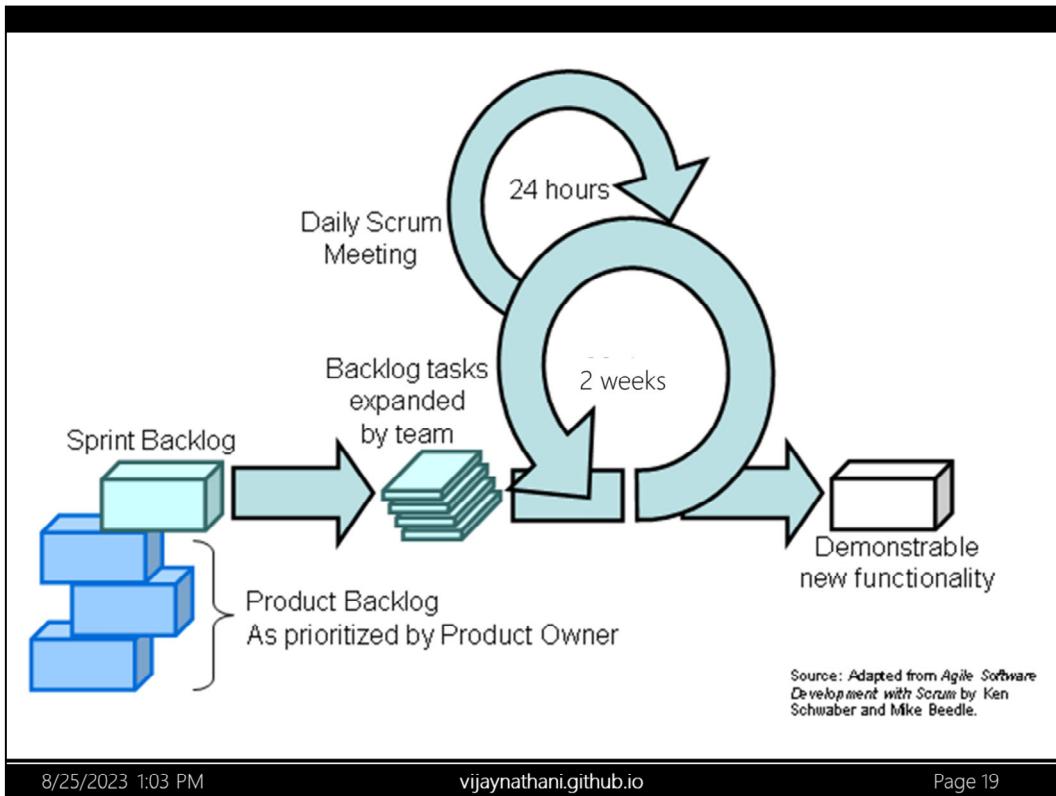
No plan survives contact with the enemy.

Iterative Development



Incremental Development





8/25/2023 1:03 PM

vijaynathani.github.io

Page 19

Who is using Scrum?

- Independent Software Vendors (ISVs)
- Fortune 100 companies
- Small startups
- Internal development
- Contract development

Who Is Using Scrum?

Small startups to large corporations:

- Yahoo!
- Sun
- Cisco
- Nokia
- Philips
- BBC
- IBM
- US Federal Reserve
- SAP
- HP
- Motorola
- TransUnion
- Google
- Microsoft

Longer list at: scrumalliance.pbwiki.com/Firms+Using+Scrum

8/25/2023 1:03 PM

vijaynathani.github.io

Page 21

IBM & Microsoft (USA) are in the process of adopting Scrum.

Some case studies

<http://www.artima.com/forums/flat.jsp?forum=155&thread=241943>

100% Scrum

PatientKeeper, Solutions IQ, Trifork,
Systematic Software Engineering,
Saxo Bank, Xebia, Innova Solutions

Partial Scrum: Microsoft, Google, Yahoo, Adobe, GE, Oracle, Siemens, Bellsouth, Ulticom, Palm, St. Jude Medical, Digichart, Healthwise, Sony/Ericsson, Accenture, GuideWorks, Exigen Services / Starsoft labs, SirsiDynix, Softhouse, Phillips, Barclay Global Investors, Constant Contact, Wellogic, Innova solutions, Medco, Myspace, Johns Hopkins, Wachovia, IBS Interprit, Openview Venture Partners.

What Is Scrum Being Used For?

- US FDA-approved software for X-Rays, MRIs
- High availability systems (99.9999% uptime)
- Enterprise workflow systems
- Financial payment applications
- Large database applications
- Embedded systems
- ISO 9001 organizations
- CMMi Level 5 organizations
- Onshore / offshore development

Is Scrum Scalable?

- Typical Scrum team is 5-10 people
- Sutherland used Scrum in groups of 600+
- Mike Cohn used in groups 100+

Microsoft and Cisco, each has Scrum team of more than 200 people.

The largest project that has used scrum had 3500 people
and lasted for 2.5 years

Eclipse has hundreds of developers spread all over the world. It is using agile process.

IBM has more than 30,000 employees. Some of its projects are agile. Some are not yet agile. It is in process of transition.

ibm.com uses Scrum and XP.

Scrum Characteristics

- One of the “agile processes”
- Self-organizing teams
- Product progresses in a series of “sprints”
- Requirements are captured as items in a list of “product backlog”
- No specific engineering practices prescribed
- Uses generative rules to create an agile environment for delivering projects

What is Scrum?

- There is a simple, Boolean test for whether a project is practicing Scrum
 - Scrum has specific roles
 - Scrum has specific practices
 - Scrum has specific artifacts

Everything else is not part of Scrum

Scrum is a Mindset

- Scrum is commitment-oriented.
- Scrum is results-oriented: projects produce increments of a shippable product, activities are time boxed, and ceremony is discouraged.
- Scrum is disciplined. There are practices you must follow on a specified time table.



Scrum Roles

- The Product Owner
- The Scrum Master
- The Team
- Everyone else is not part of Scrum



Scrum Practices

- The Sprint Planning Meeting
- The Sprint
- The Sprint Review Meeting
- The Sprint Retrospective
- The Daily Scrum
- All other practices are not part of Scrum

Scrum Artifacts

- The Product Backlog
- The Sprint Backlog
- The Sprint Burndown Chart
- The Product Increment
- Everything else is not part of Scrum

For Empirical Process

- Simple, clear purpose and principles give rise to complex, intelligent behavior.
- Complex rules and regulations give rise to simple, stupid behavior
- - (Hock 1994).

8/25/2023 1:03 PM

vijaynathani.github.io

Page 31

PMBOK guide is hundreds of pages.

PMI PMP certification says:

1. Identify the requirements of the project
2. Establish objectives that can be achieved
3. Balance scope, cost and time
4. Satisfy everyone's needs

It says monitor and control the project.

Process groups: Initiating process group, Planning process group, Executing process group, Monitoring&Controlling process group, Closing process group.

44 processes: Develop project charter, Develop preliminary scope statement, Develop Project Management Plan, Risk identification, Qualitative Risk Analysis, Quantitative Risk Analysis, Risk response planning, Quality planning, Activity Duration Estimating, Direct and Manage work, Information Distribution, Scope control, Monitor and control project work, schedule control, close project ,etc.

Nine knowledge areas: Risk, Quality, Scope, Procurement, Communication, Time, Integration, Cost, HR.

For Time Management PMBOK says: Define activities, Sequence them, Estimate resources, Estimate duration, Schedule development, control Schedule.

It uses critical path, slack, resource leveling, leads and lags, schedule compression techniques

PMBOK says monitor costs by estimating, budgeting and controlling.

PMBOK says Quality is the measurement of how closely your product meets its requirements.

PMI has complex rules and regulations. It leads to the stupid behavior of cutting quality for the deadline.

Scrum is simple and lets people use their intelligence to decide the priorities. Because the process is transparent, Quality is rarely cut.

Module: Scrum Basics

"In the long history of humankind (and animal kind, too), those who learned to collaborate and improvise most effectively have prevailed." — Charles Darwin

Scrum Roles – Brief Introduction



Product Owner

Determines what should be produced, to maximize ROI for the business



The Team

Cross-functional and self-organizing team of 7 people +/- 2



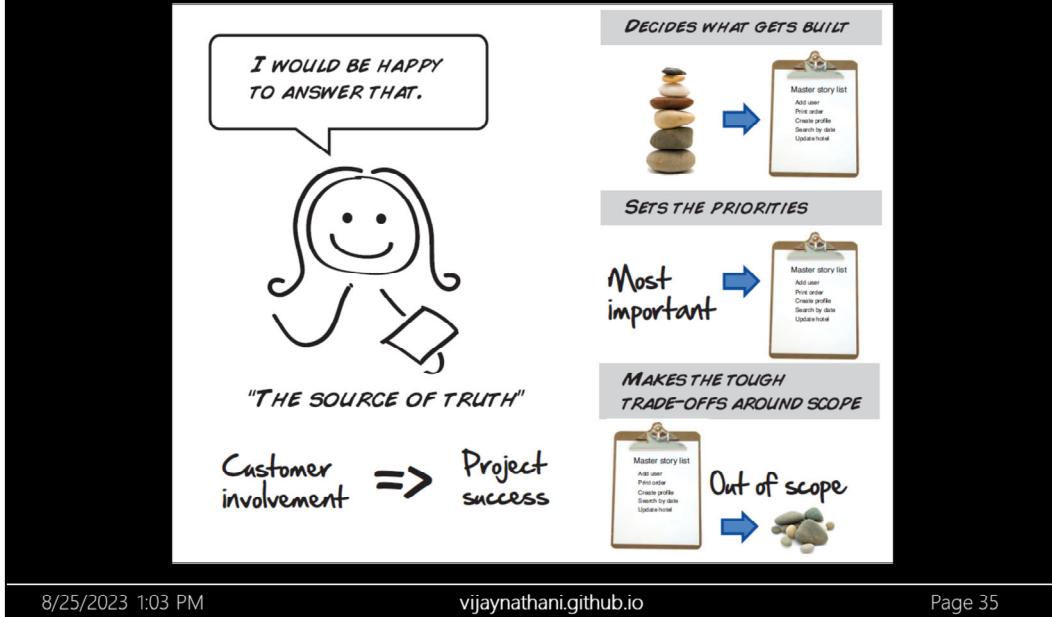
ScrumMaster

Protects and serves the Team, and supports their practice of Scrum

The Basics of Scrum



Product Owner



8/25/2023 1:03 PM

vijaynathani.github.io

Page 35

Owns definition of success

Charged with maximizing ROI and managing project risk

Responsible for taking all inputs into what the team should produce, and turning it into a prioritized list. Inputs include:

Customer

Team

Executives

Competitors

Other Stakeholders

Determines release plan and communicates it to upper management and the customer

=

One study found that all the bugs that developers are responsible for, in 37% of the cases, they just did not know that they were supposed to do that check. Systems are so complex today, that one person rarely knows the entire system. So closely working with the PO is important. If possible, the PO should sit in the same areas as the team.

Responsible for all Features

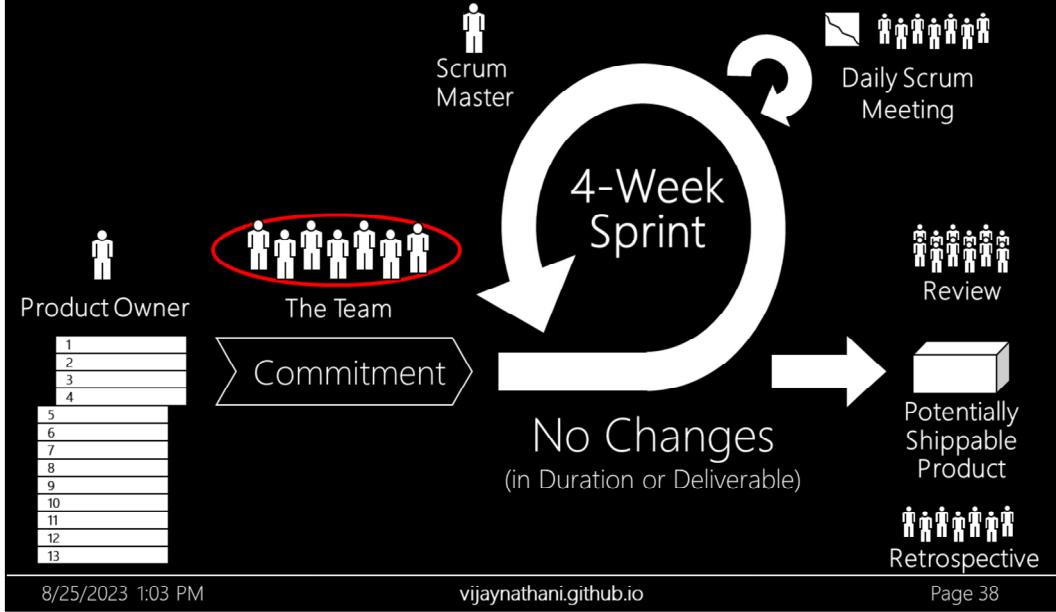


There is one Product Owner. Only one.

Product Owner

- Represents all stakeholders
- Decides **where** the team should go.
 - Not how they should get there
 - Not their speed.
- Defines scope. Prioritizes.
- Owns the product backlog
- Not a line manager.
- Does not estimate stories

The Basics of Scrum



The power of We is stronger than the power of Me.

A team is a small number of people with complementary skills who are committed to a common purpose, performance goals, and approach for which they hold themselves mutually accountable.

What is a team

- Teams are usually small – 5 to 10 members
- A team has a common purpose
- Team members make commitments about tasks to each other
- Teams has peers



8/25/2023 1:03 PM

vijaynathani.github.io

Page 39

A Manager are not a part of developer team

The structure of a team is network and not hierarchy

Teams in Agile Ecosystems

Some people think that "Agile" means

fewer processes,

less ceremony, and

briefer documents,

but it has a much broader perspective.

Although fewer processes and less formality might lower development costs, they are not enough to produce agility.

Focusing on people and their interactions and giving individuals the power to make quick decisions and to self-adapt their own processes are key to Agility.

The Team

- “The Team decides how to turn the selected requirements into an increment of potentially shippable product functionality. The Team devises its own tasks and figures out who will do them.”
- Owns the production and engineering process
- Meeting all goals within the guidelines, standards, and conventions of the organization and of Scrum.

8/25/2023 1:03 PM

vijaynathani.github.io

Page 40

Dee Huck said,

“Simple, clear purpose and principles give rise to complex, intelligent behavior”

“Complex rules and regulations give rise to simple Stupid behavior”

Self Organizing Teams are Superior to Command n' Control Teams

* People in a self-organized team are able to make decisions themselves and accordingly adapt to changing situations. Command and control grunts have to wait for the boss to tell them what to do. That introduces latency in the development process as the team waits for the leader to shake free to deal with a decision. It can also sap a developer of any energy to contribute to the design and approach if they know they don't have any say in the matter. It's a negative incentive to increase their intellectual participation in the project, and that can only hurt the team.

* Self organized teams do a much better job of utilizing the talents of the team because more minds are involved in any activity. By investing one person with all decision making authority, you effectively shut down the other team member's contributions to the design and planning. They become drones.

* Self organized teams have much more communication between team members. A command and control team chokes communication by running too much through one person.

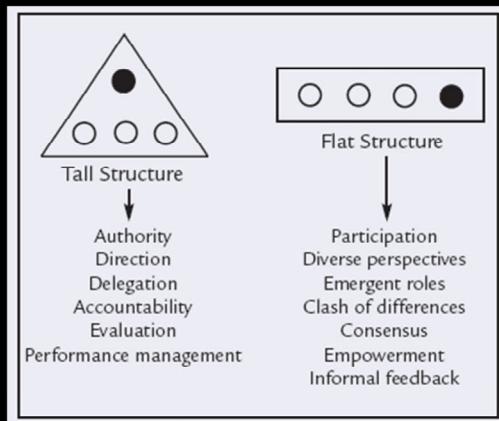
* Command and control organizations don't provide as many chances for personal development. The best way to learn is to have actual responsibility and opportunities to do new things. If all you do is what you're told without question, you don't get to learn how to make decisions.

* A self organized team is collectively aware of the upcoming work and much better able to bootstrap themselves with new work when they complete their existing task. I've learned that there's a lag between telling a developer to write a feature and the beginning of coding because it takes some time to understand what's required. You can cut down that lag by doing planning and design together as a team.

* Self organized teams spread knowledge around much better and make decisions together. That makes each team member more effective because they have much more background on the "why" of the coding assignments. A command and control team member often lacks an understanding of why a decision was made because they weren't involved with that decision. That hampers their ability to follow a design or approach.

Team

- Cross-functional
- Self-organizing
- Has worked with as high as 15, as few as 3



8/25/2023 1:03 PM

vijaynathani.github.io

Page 41

Cross –Functional: (QA, UI designer, Programmers, etc)

Possesses all the skills necessary to produce an increment of potentially shippable product

Empowered to do whatever's necessary to produce the potentially shippable product, within the constraints of the organization's standards

Team:

Team members should be full-time.

If a team-member is part-time, treat the person as a consultant i.e. the person will report to the team. He/She does not have to attend daily meeting or any other meeting.

Can be shared with other teams (but better when not)

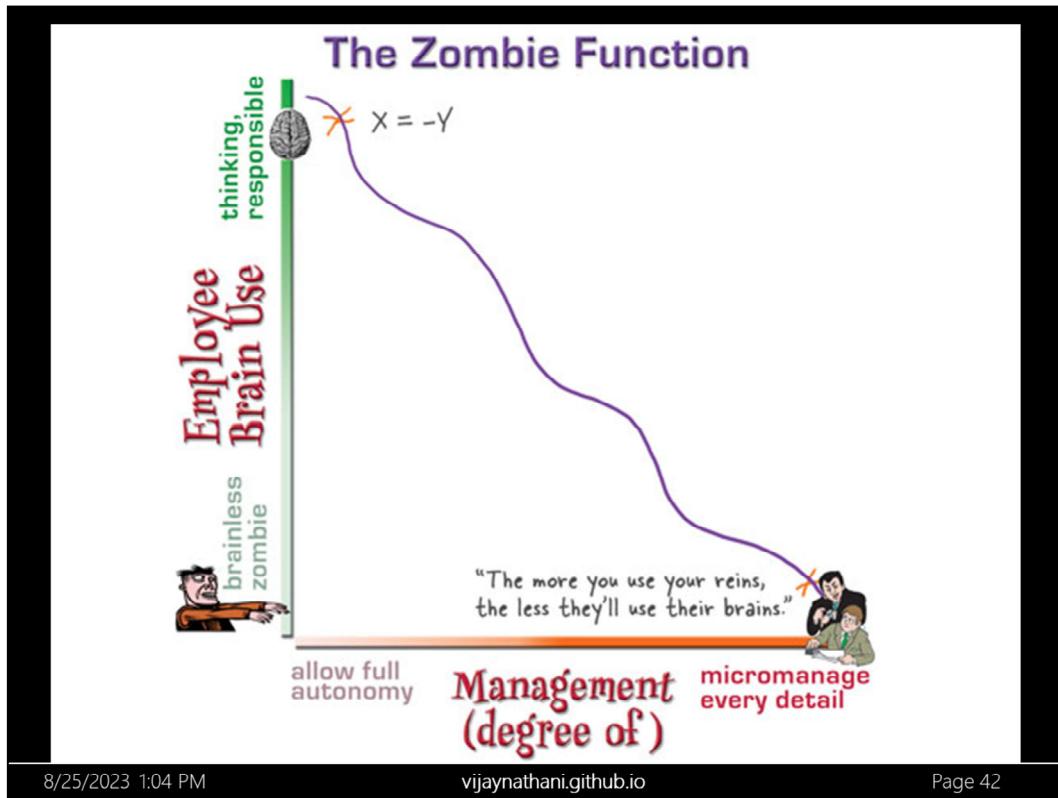
Can change between Sprints (but better when they don't)

Can be distributed (but better when colocated)

=====

Self-organization does not mean without flow, or form, or context. Nor does it mean leaderless. To foster self-organization means to:

- Encourage teams to figure out the best way to complete the defined work.
- Encourage decision making fit for the occasion.
- Remind teams to use their charter as their guide.



If people are not empowered to make improvements, they stop thinking about making improvements.

The team is both empowered and accountable to deliver the goods. The team does their job when they self-organize, self-manage and self-achieve the objectives of the Sprint.

For many organizations, this turns things upside down.

The hierarchical-technical-management-directive approach is essentially eliminated with Scrum.

The Product Owner now sets the objectives and priorities, the team figures out how to achieve them, and no one need tell them how to do that along the way.

Software Development is a social Activity

If you want to be incrementally better: Be competitive. If you want to be exponentially better: Be cooperative.

A strong player is not necessarily an ace programmer. A strong player may be an average programmer, but someone who works well with others.

Working well with others, communicating and interacting, is more important than raw programming talent. A team of average programmers who communicate well are more likely to succeed than a group of

superstars who fail to interact as a team.

It isn't what you don't know that gets you in trouble. It's what you know that isn't so. The biggest problem, in what people "just know" about software development, is that they are focused on individual action. What actually matters is not how any given person behaves, but how the individuals behave as part of a team.

Good Teams



8/25/2023 1:04 PM

vijaynathani.github.io

Page 43

The catalyst is important because a project is always in a state of flux. Someone who

can help a project to jell is worth two people, who just do work.

=====

Sport is not a good analogy for Agile team because the extra player is hoping that one player breaks his leg so that he can play.

In Music group, either the band succeeds or it fails. One person cannot succeed while the others fail.

=====

Basic truths about team performance:

1. Teams and people do their best work when they aren't interrupted;
 2. Teams improve most when they solve their own problems; and,
 3. Broad-band, fact-to-face communications is the most productive way for teams to work together.
- =====

Basic truths about team composition

1. Teams are more productive than the same number of individuals;
2. The optimum size team is around seven people, and no more than nine;
3. Products are more robust when a team has all of the cross-functional skills focused on the work; and,

4.Changes in team composition ruin productivity.

Co-operative or Competitive

- Co-operative, Goal seeking, Load bearing, Team, Individual with talent, Skill-sensitive, Training, Tools, Resource limited, Plan, Improvised, Fun, Challenging, Dangerous



8/25/2023 1:04 PM

vijaynathani.github.io

Page 44

Games are either co-operative / competitive

In cooperative games, the people work either to win together or to continue the game as long as they consider it worth playing. Storytelling, playing jazz, etc are non-goal-seeking cooperative games. In these latter games, the players do not seek to end the game by reaching a goal as fast as possible. They come to an end only when enough people get tired of playing and step out.

Cooperative and goal-seeking. A team of rock climbers work together to reach the top. They will evaluate the climb based on how well they climbed together and how much they enjoyed themselves, but the first measure of success is whether they reached the top. Reaching the endpoint is a primary goal, and the game is over when they reach the top.

(If you are a rock climber, you might well interrupt me here. For many rock climbers, the moment of reaching the end of the climb is a sad one, for it signals the end of the game. That is true of cooperative games in general. The game comes to an end when the endpoint is reached, but if the players have been enjoying themselves, they may not want to stop. Similarly, sometimes software developers do not want to finish their design, because then the fun part of their work will be over.)

Load bearing. The climbers must actually support their weight on their hands and feet. This is a particularly valuable point of comparison between the two: Software must run and produce reasonable responses. While multiple solutions are possible, not just any solution will do.

Team. Climbing is usually done in teams. There are solo climbers, but under normal circumstances, climbers form a team for the purpose of a climb.

Individuals with talent. Some people just naturally climb better than others do. Some people will never handle certain climbs.

Skill-sensitive. The rock climber must have a certain proficiency. The novice can approach only simple climbs. With practice, the climber can attack more and more difficult climbs.

Training. Rock climbers are continually training on techniques to use.

Tools. Tools are a requirement for serious rock climbing: chalk, chucks, harness, rope, carabineer, and so on. It is important to be able to reach for the right tool at the right moment. It is possible to climb very small distances with no tools. The longer the climb, however, the more critical the tool selection is.

Resource-limited. A climb usually needs to be completed by nightfall or before the weather changes. Climbers plan their climbs to fit their time and energy budget.

Plan. Whether bouldering, doing a single-rope climb, or doing a multiple-day climb, the climbers always make a plan. The longer the climb, the more extensive the plan must be, even though the team knows that the plan will be insufficient and even wrong in places.

Improvised. Unforeseen, unforeseeable, and purely chance obstacles are certain to show up on even the most meticulously planned climbing expeditions unless the climb is short and the people have already done it several times before.

Therefore, the climbers must be prepared to change their plans to improvise at a moment's notice.

Fun. Climbers climb because it is fun. Climbers experience a sense of flow while climbing, and this total occupation is part of what makes it fun. Similarly, programmers typically enjoy their work, and part of that enjoyment is getting into the flow of designing or programming. Flow in the case of rock climbing is both physical and mental. Flow in the case of programming is purely mental.

Challenging. Climbers climb because there is a challenge: Can they really make it to the top? Programmers often crave

this challenge, too. If programmers do not find their assignment challenging, they may quit or start embellishing the system with design elements they find challenging (rather like some of the poets mentioned in the epic poetry project).

Dangerous. Probably the one aspect of rock climbing that does not transfer to software development is danger. If you take a bad fall, you can die. Rock climbers are fond of saying that climbing with proper care is less dangerous than driving a car. However, I have not heard programmers express the need to compare the danger of programming with the danger of driving a car.

Software development has been compared with many other things, including math, science, engineering, theater, bridge building, and law. Although one can gain insight from looking at any of those activities, the rock-climbing comparison is the most useful for the purpose of understanding the factors involved in the activity.

=====

Rock climbing has one limitation when used as a metaphor for software development: When people finish a rock climb, they walk back down to their cars and drive off. When people finish a software project, they stay around and enhance the system or build the neighboring system. They have to live with the results of what they did, and they have to integrate other people into the world they created.

Make a choice

Predict or Adapt—you can't have it both ways.



8/25/2023 1:04 PM

vijaynathani.github.io

Page 45

Project managers who believe in predictability will control projects by conformance to plan.

Being Agile and adaptable isn't about practices, it's about deep-down fundamental philosophy - A healthy emphasis on the virtues of teamwork in a business plagued with prima donnas.

Prima Donna - conceited, difficult person: a person who is regarded as demanding and difficult to please (*insult*)

Project managers who believe in adaptability will control projects by demonstrating results and measuring business value along the way. Agile developers and managers have more confidence in their ability to adapt than in their ability to plan.

Team Member's Role

- Accept responsibility for results
- Accept responsibility for relationships (respect, trust, etc)
- Confront reality through rigorous thinking
- Engage in intense interaction & debate

He who passively accepts evil is as much involved in it as he who helps to perpetuate it. – Martin Luther King Jr.

8/25/2023 1:04 PM

vijaynathani.github.io

Page 46

Once a team begins to jell, the probability of success goes way up. It doesn't need

to be managed in a traditional way, and they certainly don't need to be motivated.
It has momentum.

Team members must have
trust in one another.

Mavericks may have to be
excluded from the team

A team gains consensus



- All members commit
- Decisions are made highly visible
- Agreement is through consensus

- Check on consensus to quickly find the disagreements
- Fist of five approach:
 - 5 = wild, unbridled support
 - 4 = this is a fine idea, wish I'd thought of it
 - 3 = I can live with and support it
 - 2 = I have reservations I'd like to think about
 - 1 = I am very opposed; we shouldn't

8/25/2023 1:04 PM

vijaynathani.github.io

Page 47

The IQ of the crowds is better than the IQ of a single person.

“Wisdom of Crowds” book:

- “Who wants to be millionaire?” show offered the participants 3 choices: 50-50, phone a friend, audience poll. The success of audience poll was 91 %. The success of phone a friend is 65%.
- A professor took a different jars of beans and asked the students to guess the number of beans. The average consistently was better. The 1 / 2 person who performed better than the average could not retain their correctness. So we score better by going with the average, rather than one single person.

For this approach to work, each person must think independently. A person must not take decision based on the opinion of others.

e.g. Two equal lines were drawn on the board. If a large number of (biased) people said that one is bigger than the other, the unbiased participant usually said the same. This defeats the purpose of independent thinking.