

# AGILE, IMPROVEMENT, EMPOWERMENT

How Agile Improves Team Empowerment

## TDS AGILE GROWTH

- Agile Growth is a team in TDS, centered in the PMO, that focuses on Agile coaching and guidance to help teams.
- The group also analyzes processes and process improvement.

## WHO WE ARE

- Steven Savage – Lead, embedded coach
- Brendan Murphy – Events expert, coach
- Michale McCaulliffe – Education and materials
- Anil Kumar - Standards
- Caroline Hsia – Infrastructure expert
- Richard Wiecki – Finance expert, embedded Agilist

# AN INTRO TO AGILE

# PROJECT APIS

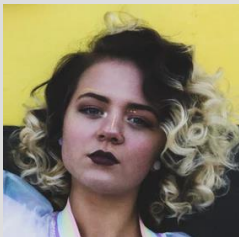
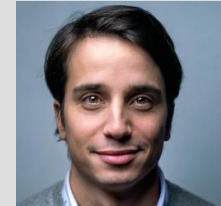
- A web-based app to connect students, mentors, and projects for medical research and education.
- Designed to be rapidly developed and have mobile components.
- Mixes old and new technology.
- Current status: “on fire and we’re arguing with each other.” (aka “Red but we’re calling it Yellow”)

## MEET THE TEAM



“I’m Ruth, the Analyst. I determine what the users needs and how the software should work. But people change their minds, the requirements are confusing, and no one wants to say what the problems are.”

“I’m Sai, the Project Manager. I help plan the projects and guide the teams to get things done. Our time estimates are always off – even with the hard work. I hate giving status reports, and because there’s too many unknowns.”



“I’m Kel, the lead Programmer. I help the team solve problems, plan, and analyze their approaches. We’re working overtime, not getting things done, the requirements aren’t what people say they are. Don’t start me on the technology!”

## AGILE – PEOPLE FOCUSED

- Each person on a project knows their job.
- Each person on a project has a useful perspective.
- Most problems aren't due to people – it's that people need to be supported, listened to.
- What if we were to think about the people involved?  
**That's Agile.**

# PHILOSOPHY - THE AGILE MANIFESTO



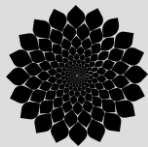
**Individuals and interactions over processes and tools**



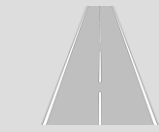
**Working software over comprehensive documentation**



**Customer collaboration over contract negotiation**



**Responding to change over following a plan**





## AGILE IS ABOUT BEING WHERE YOU ARE

- Agile is about transparency and honesty.
- You admit where you are.
- You talk, you negotiate.
- You adapt.
- You deliver incrementally to learn.
- **You accept other people where they are and support them.**

## QUESTION #1: WHAT IS YOUR AGILE STRENGTH?

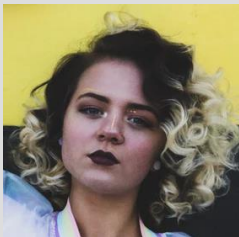
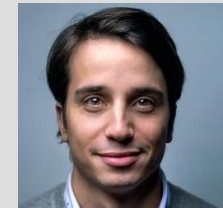
- **Communicating with people!**
- **Delivering the product no matter what!**
- **Collaborating!**
- **Responding to change!**

## THE APIS TEAM REACTS TO AGILE



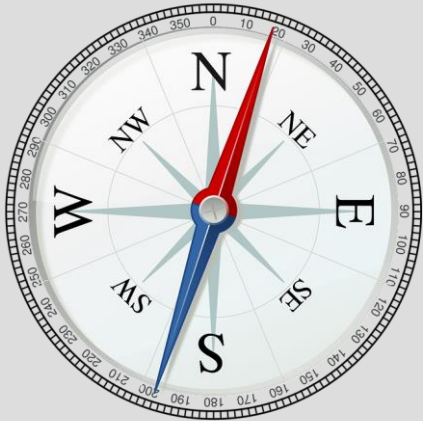
(Ruth, the Analyst): I like the focus on being able to change. But I'm wondering how we get people to stop arguing tiny details and really collaborate. Everything feels like a fight – does anyone **want** to do this project?

(Sai, the Project Manager): People talking to each other more is exactly what we need. But I don't see how that helps my team plan or accomplish anything. Where's the **plan** – because people want to see that even if it fails.

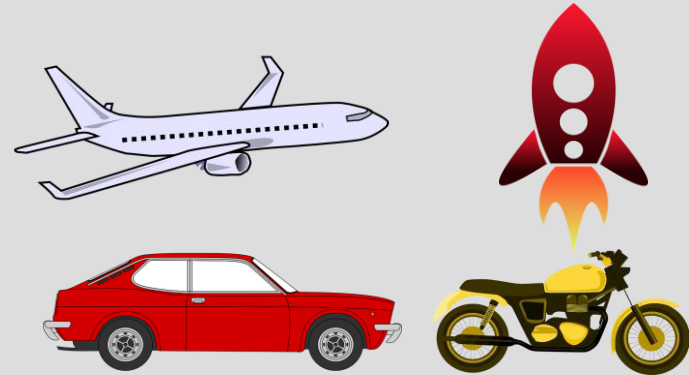


(Kel, the Lead Programmer): Getting rid of too many tools and focusing on making the product useful? Sign me up - let me guess, the methods help us get there? Because I've heard "we're going to fix this" before and **no one did**.

# MAKING IT WORK



Agile is a **Philosophy** about project productivity that focuses on *iterative development* where *solutions evolve by collaboration*.



Agile contains **Methods** to bring Agility about. They help you get to your destination different ways depending on your needs.

## THERE ARE MANY AGILE METHODS FOR TEAMS TO CHOOSE!

- Scrum – Iterative phases and clear roles.
- Kanban – Constant, gradual work.
- Disciplined Agile Development – Prescriptive approaches.
- Extreme Programming – Maximize what works.
- ...And more.

**Already on top of this.** I started reading up on the internet as soon you mentioned this. Scrum sounds perfect. And, yes, I already asked the team.



# EXAMINING AN AGILE METHOD

Scrum As Solution And Example

# SCRUM - OVERVIEW



## THE TEAM

- Product Owner – Knows Product
- Scrum Master – Coaches, guides
- Team – Makes product



## THE CYCLE

- Regular Sprints to do work
- Each Sprint take topmost work to get done
- Complete Sprint and deliver
- Adjust and plan next Sprint



## THE BACKLOG

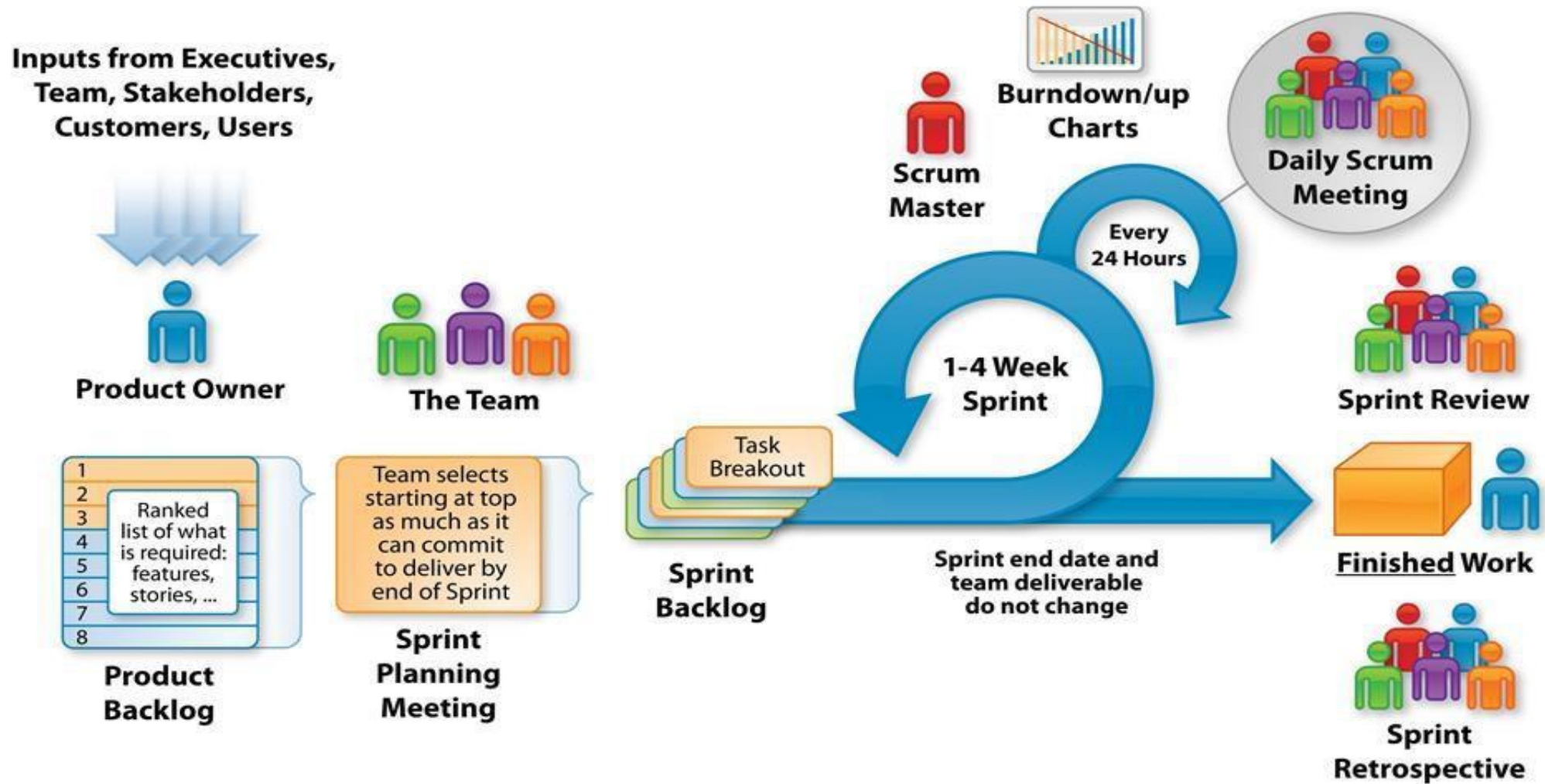
- Created by Product Owner
- Small, discreet stories in order of importance
- Groomed constantly



## THE EVENTS

- Planning Session
- Daily Standup
- Demonstrations
- Retrospectives

# The Agile - Scrum Framework





## SCRUM EMPOWERS PEOPLE

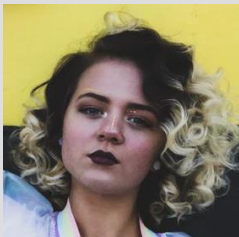
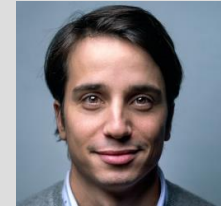
- Supports communication.
- Short phases for feedback.
- Backlogs are based on collaboration, with details discovered as needed.
- A process-and-tool-light yet organized approach.
- Change is worked into the method.

## THE TEAM REACTS



(Ruth, the Analyst) I like how change is baked into the method, and it's very, very transparent. It sounds like we won't miss anything – and it's honest.

(Sai, the Project Manager) It's a plan that's also not a plan, and statuses are on one board and from just having meetings. I'm for it. Everyone will know what's going on – and I'll be less of a middleman.



(Kel, the lead Programmer) Pacing ourselves is something we need. Also maybe all of us collaborating mean changes will be easier, since we won't get surprised. Also people will be listening to engineers now. That'll be new.

## SCRUM SHOWS HOW AGILE EMPOWERS

- A team approach. The distinct roles just provide clarity.
- Transparency and honesty builds trust.
- Collaborative – people work together to solve a problem.
- Provides an easy framework to work in.

## QUESTION #2: WHICH PART OF SCRUM INTERESTS YOU?

- **Clear Roles**
- **One ordered backlog**
- **Short sprints with deliverables**
- **Daily checkins**
- **Pre-planned meetings**
- **Review of results**

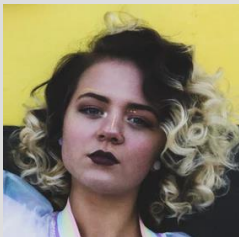
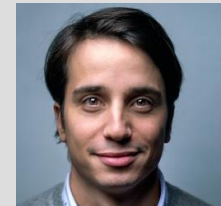
**GETTING GOING**

# CHOOSE THE ROLES



(Ruth, the Product Owner ~~Analyst~~) I'll be Product Owner. I know everyone anyway, and it lets me focus on getting a good backlog. I also take some of Sai's duties on reporting.

(Sai, the Scrum Master ~~Project Manager~~) I'm the Scrum Master. I've been running events anyway, and I've been informally coaching the team. This lets me focus on helping the team get the job done instead of a bunch of status reports.

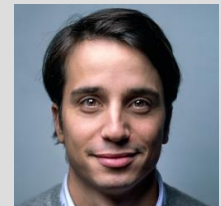


(Kel, the lead Programmer) Honestly, my job doesn't change much except I have less responsibilities. We expect the team to collaborate. Maybe I can actually do some programming ...

## BUT NOW WE NEED A BACKLOG!

- Normally, you could have the Product Owner make one, but this is a continuing project.
- Also, you need the teams to get used to working together.
- There are many methods, but ...

(Sai, the Scrum Master) I read about this! In my role as Scrum Master, I will “coach” people to use a Sprint Zero! Let’s get everyone together and make a backlog!



## AN AGILE TECHNIQUE - SPRINT ZERO

- Everyone involved gathers to outline all the work done.
- It's not a plan – it's **enough work to get done**
- It's also done **together** so engineers, end users, and more are involved.
- This could be a day – or it can be up to two weeks.



(Ruth, the Product Owner) Everyone decided I should handle this – and I agree. I know the people and the product anyway. But me, Sai, and Kel will stay out of the way as much as possible. Agile is about team effort.



# PLANNING

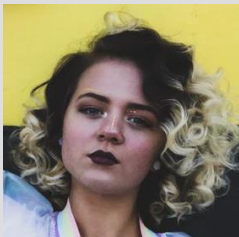
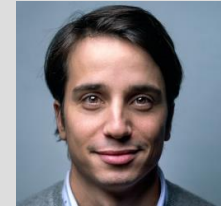
Selecting our work

## THE SPRINT STARTED! HOW WAS PLANNING?



(Ruth, the Product Owner) The end users and I had to answer a **lot** of questions. It made us think, and we had to change priorities once we talked to the programmers. The end users seem to understand the big picture now.

(Sai, the Scrum Master) It took awhile. People grumbled, but they got questions answered, and it's only every two weeks. I already have ideas to help it go faster – I've been reading.



(Kel, the lead Programmer) I can't believe we spent four hours on this, but also I think we have a workload we can do that people want done for the first time. I just hope we can speed it up, or at least get a free lunch.

## AGILE PLANNING IS COOPERATIVE

- All Agile Methods have cooperative planning.
- Everyone knows something – and everyone should be listened to.
- Be inclusive – you never know who has something to contribute.



(Ruth, the Product Owner) I may own the product backlog, but the product belongs to everyone. Kel was surprised that I invited the training lead, but she showed how a slight schedule change let us finish training videos earlier.

# DAY BY DAY

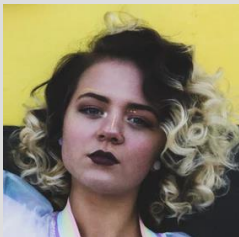
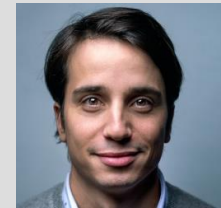
What's every day like?

## WHAT'S EACH DAY LIKE



(Ruth, the Product Owner) A lot more human contact for me – and that's saying something. I spend a lot of time answering questions, connecting people up, and grooming the backlog. Glad I could offload some duties!

(Sai, the Scrum Master) More predictable for me. I always know what's coming, the daily standups keep us in touch. Though now I'm spending more time troubleshooting problems for the team so they focus – I'm learning a lot.

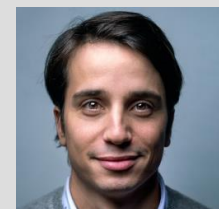


(Kel, the lead Programmer) We have time to focus and having daily touchbases is a big help. Also, we're hearing from end users a lot more. Plus any blockers we just toss to Sai or Ruth so we keep going.

## PEOPLE COMMUNICATE REGULARLY

- Scrum has formal meetings to ensure communications.
- ALL Agile methods support and encourage communication.
- People need to be coached to support communications – and listened to.

(Sai, the Scrum Master) Now that I can coach people, I'm helping them communicate better. A lot of it is listening or making space, but here's some things I found ...



## AGILE COMMUNICATION TWEAKS

- “Use It” – Use existing meetings to do more.
- “War Rooms/Swarming” – To solve a problem, sometimes everyone gets in a (virtual) room and focuses on an issue.
- “Open Hours” – Certain people are always available at a specific time.
- “Open Space” – For a time everyone is in the same (virtual?) space, just so they can ask questions **if needed**.
- Chats and emails – With the right protocols.

# END OF SPRINT

How does it end.

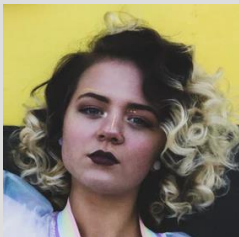
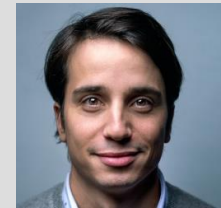


## WHAT'S THE END OF SPRINT LIKE



(Ruth, the Product Owner) The first one took nearly six hours of demonstrations, discussions, retrospectives, and planning. We made an event of it – but it was still **six hours**. Then again we got a lot done.

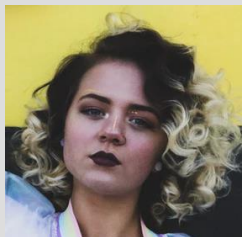
(Sai, the Scrum Master) We got 80% of what we wanted done – there were some surprises. But we warned people early, and we made progress. Thanks to the Retrospective, we know what went wrong **and** right.



(Kel, the lead Programmer) It's going to take us awhile to get used to “a few big meetings” but it went good. Best of all the end users see why some stories don't make it, and we even planned out a half-sprint ahead!

## END OF SPRINT IS ABOUT EVERYONE

- Everyone gathers to demonstrate, review.
- Learnings are shared, plans adjusted, and new plans made – together.
- Because everyone is working together, the past **and** future goals are realistic and sustainable.



(Kel, the lead Programmer) I've been reading some other methods besides Scrum. Even when you don't have sprints, demonstration, signoff, and reviewing work are part of Agile. The three of us are building a library of Agile books ...

NOW SIX MONTHS LATER

How does it end.

## QUESTION #3: HOW DO YOU DEFINE SUCCESS?

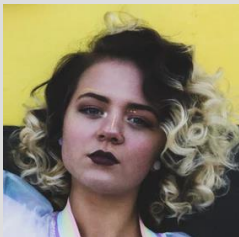
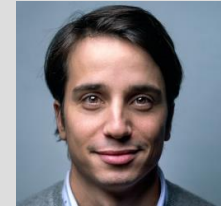
- **Getting everything we planned done**
- **Getting something that works done**
- **Getting something everyone agrees on is good enough**
- **Surviving**
- **I refuse to choose on the grounds it may incriminate me.**

## HOW DID PROJECT APIS GO?



(Ruth, the Product Owner) We didn't get all the features we wanted – but we got some the application out three months early so the end users started getting value. There's about 10-20% not done, but I don't think we needed it.

(Sai, the Scrum Master) Really good. We found a lot of challenges and unexpected problems, but just re-prioritized. Our early launch also got us a lot of feedback and helped us avoid issues.



(Kel, the lead Programmer) The team isn't burnt out now – we slowed down and set limits. We also collaborated more so the app could get out early – and there's no longer animosity.

## SUCCESS IS DELIVERING AS MUCH VALUE AS POSSIBLE IN THE BEST WAY

- **Early use** – Working together, everyone got a basic application out early. This means value delivered early and feedback.
- **Do what matters** – Do what brings the most value, and the most valuable work is done early. There is no **do it all**.
- **Sustainable** - Because realistic goals were set, the team didn't overwork themselves.
- **Cohesion** – A group that delivers better all the time.

# THE TEAM'S NEXT STEPS

Going further



## RUTH – DEVELOP MORE

I'm going to learn more about writing stories and creating backlogs. I want to help people communicate better.

Agile is an approach that helps me show the value of communication. I'm going to be pitching it to other people I work with – and Sai is studying other methods!



## THERE ARE MANY WAYS TO IMPROVE COMMUNICATION

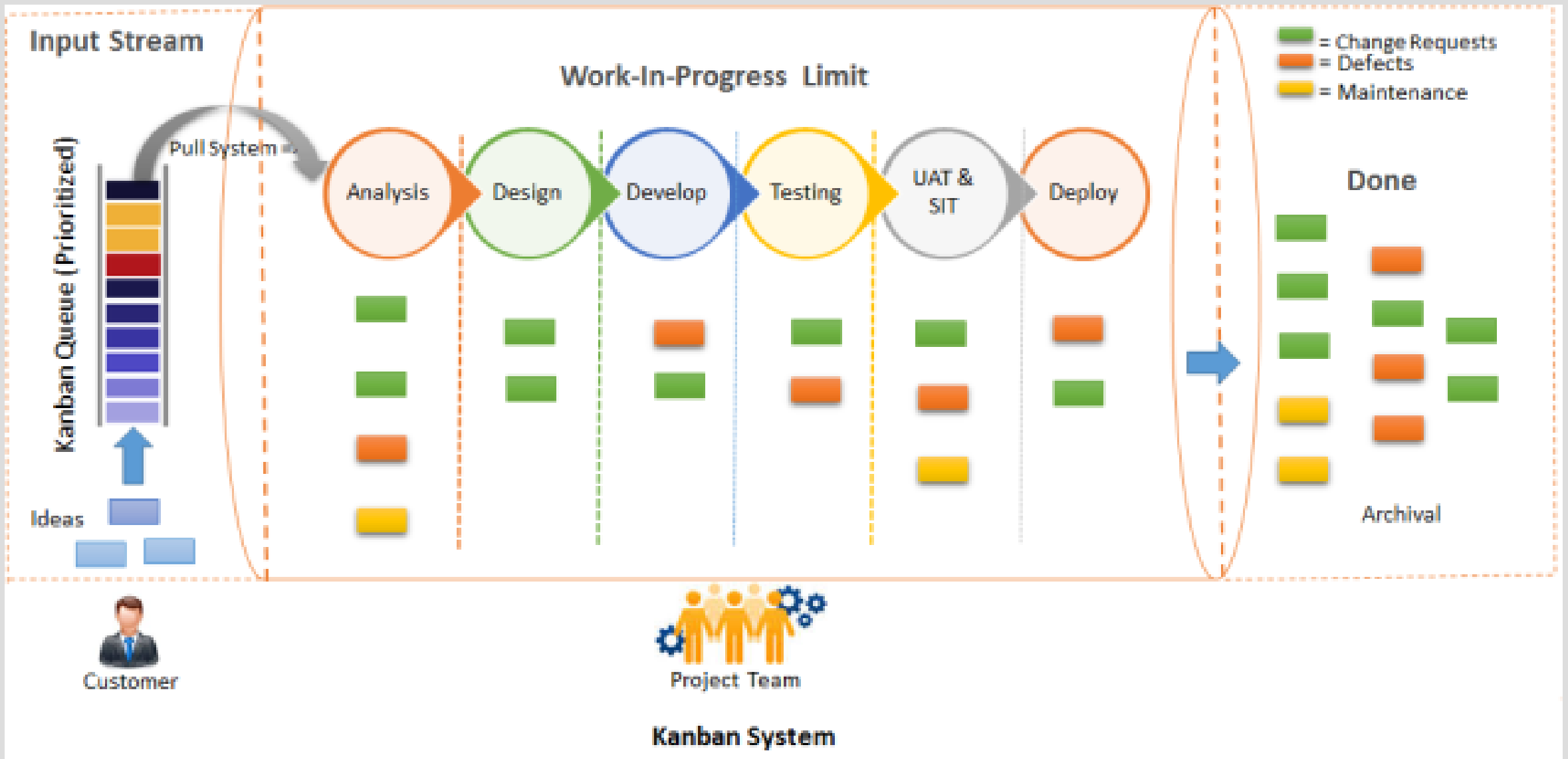
- **Story creation** - Learn how to write better stories.
- **Frameworks** – Agile methods have many story frameworks.
- **Writing** – Being able to describe things helps
- **Speaking** – Agile builds on direct communication
- **Graphics** – Infographics, charts, and more help communication.



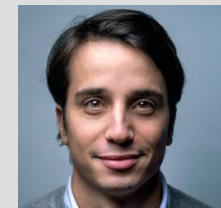
## SAI – OTHER METHODS

As Scrum Master, I'm getting certifications and studying on methods. I want to coach people to be better.

I find Agile Methods aren't one-size-fits-all, so I'm learning more. My next job may not need Scrum – in fact I really like Kanban! Let me explain why ...



(Sai, the Scrum Master) Kanban is about flow, not bounded like Scrum. It embodies Agile, and I think it's going to be good for support teams and rapid development.

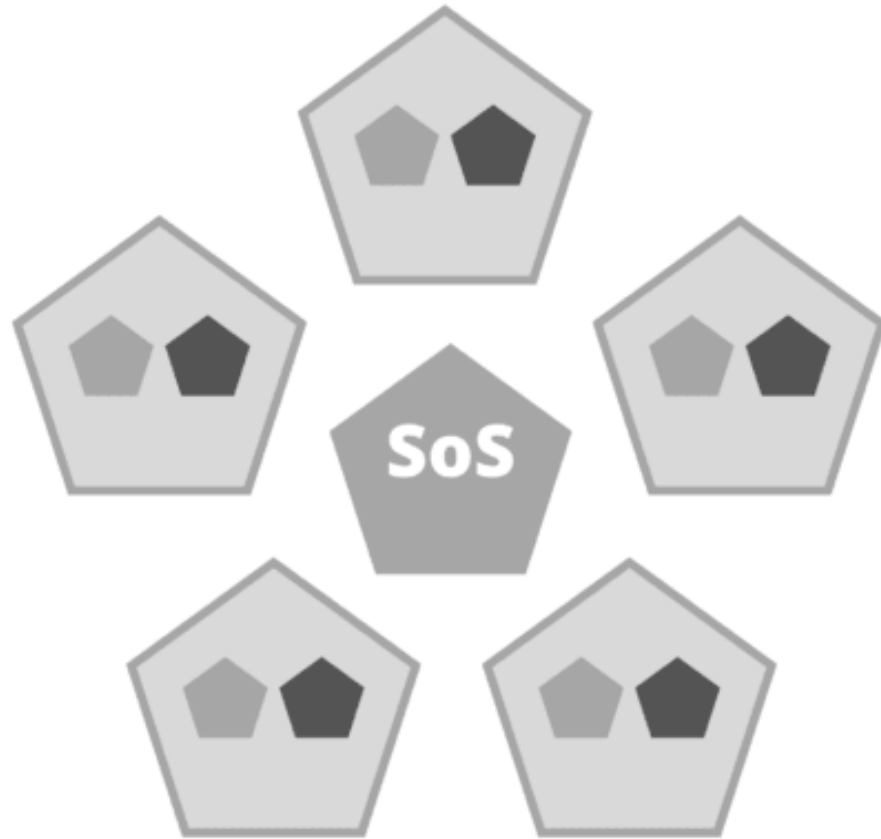




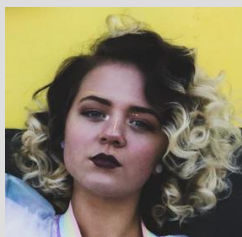
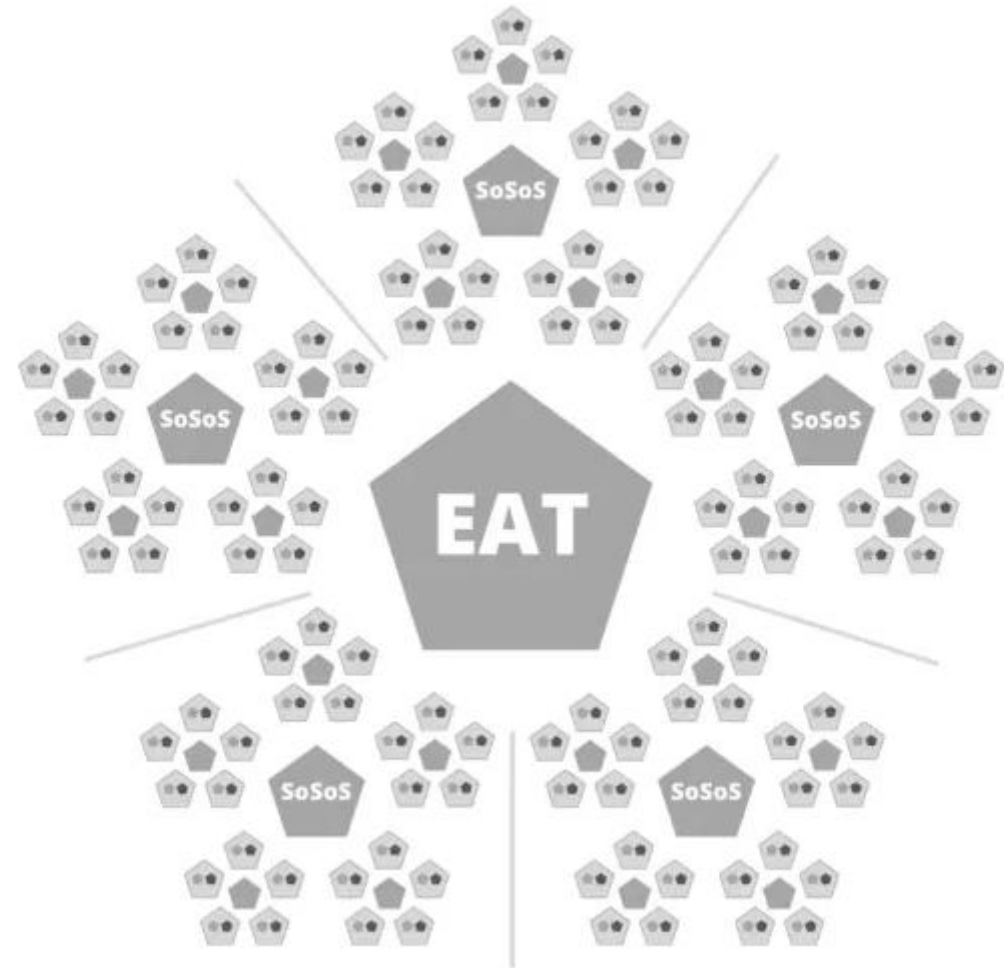
## KEL'S – THINK BIG

I've been a Lead Programmer for a long time. I'm thinking of becoming an Architect. That made me ask about ways to make Agile "Bigger."

I just found out about "Scaled Agile" and what's called "Scrum At Scale."



**SoS of 5 Teams**



(Kel, the lead Programmer) Scrum at Scale “stacks” Scrum, reporting problems all the way up to executives. It’s about coordination and solving problems! It’d be great to use when I’m an Architect!

## QUESTION #4: WHAT INTERESTS YOU IN AGILE?

- **Ways to do projects**
- **Ways to define success**
- **Ways to improve communication**
- **Ways to improve process**
- **Ways to scale our work**

# LOOKING BACK

A quick review

## AGILE EMPOWERS TEAMS . . .

- It is a **Philosophy** of ideal projects that are people/communication centered.
- It has **Methods** that put the philosophy into practice.
- It is about **where you really are** and **where other people are**.



THANKS FOR YOUR TIME

What's Next

## YOUR NEXT STEPS

- **Contact the TDS Agile Growth team for more help!**
- **Get an Agile Certification.**
- **We'll provide a reading list!**

# APPENDIX: AGILE RESOURCES!

What's Next

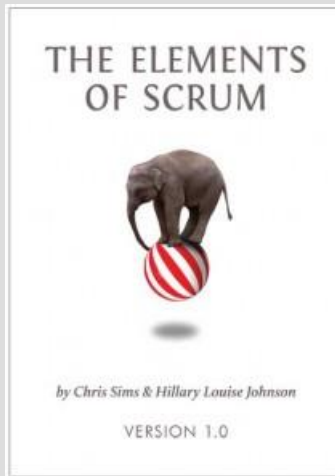
## STARTING OUT

- **The Agile Manifesto** - <https://agilemanifesto.org/>
- **The Scrum Guide** – The official, updated, scrum methods. Good ideas, even if you don't use Scrum.  
<https://www.scrum.org/resources/>
- **Kanban Guides** – Not as official as the above as Kanban has no central points, very useful anyway.  
<https://kanbanguides.org/>

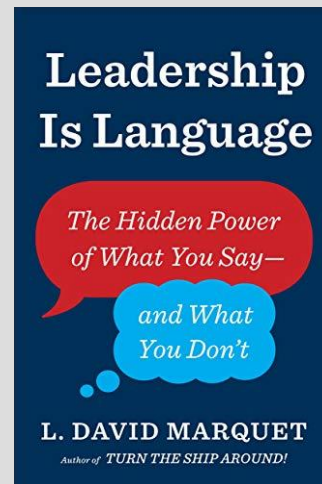
## RESOURCES

- **Starter Pack:** <http://goto.stanford.edu/agiletoolkit>
- **The TDS Talks Agile Growth Series -**  
<https://medwiki.stanford.edu/display/PIP/Series%3A+Agile+Growth>
- **Agile Stanford Mailing List -**  
<https://mailman.stanford.edu/mailman/listinfo/agile-stanford>

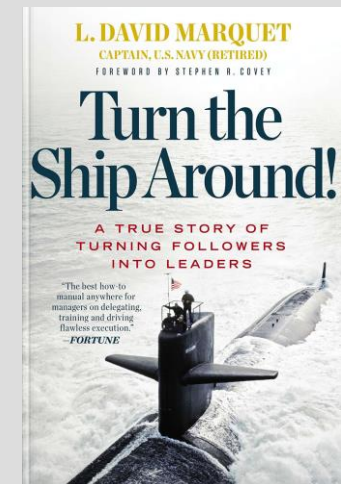
# RECOMMENDED BOOKS



[The Elements of Scrum](#)  
by Chris Sims and Hillary Johnson



[Leadership is a Language](#)  
by David Marquet



[Turn the Ship Around](#)  
by David Marquet

## SCALED AGILE:

- Scrum At Scale - <https://www.scrumatscale.com/>
- SAFE - <https://www.scaledagileframework.com/>
- LeSS - <https://less.works/>