



Buffalo WIA & Women in Tech Present:

CHOOSE YOUR OWN AGILE ADVENTURE

Discover modern ways of working by exploring the Agile Mindset

GUEST SPEAKER:
BRIAN LINK
AUTHOR & ENTERPRISE AGILE COACH

[REGISTER NOW](#)

TUESDAY
SEPTEMBER 10, 2024
FROM 5-7PM

Pearl Street Grill & Brewery
76 Pearl St
Buffalo, NY 14202

EVENT SPONSORED BY:

BUFFALO WIA & WOMEN IN TECH
WOMEN IN TECH
WESTERN NEW YORK

43NORTH Be In Buffalo

Buffalo Women in Agile + TechBuffalo Present: Choose Your Own Agile Adventure!

Brian Link

Why are we here?

Agile Team Maturity. “Being Agile”

Ways of Thinking

- An Iterative Mindset
- A Product Culture
- A Customer-Centric Mindset
- A Culture of Learning
- A Culture of Experimentation
- A Culture of Continuous Improvement
- A Culture of Psychological Safety

Ways of Working

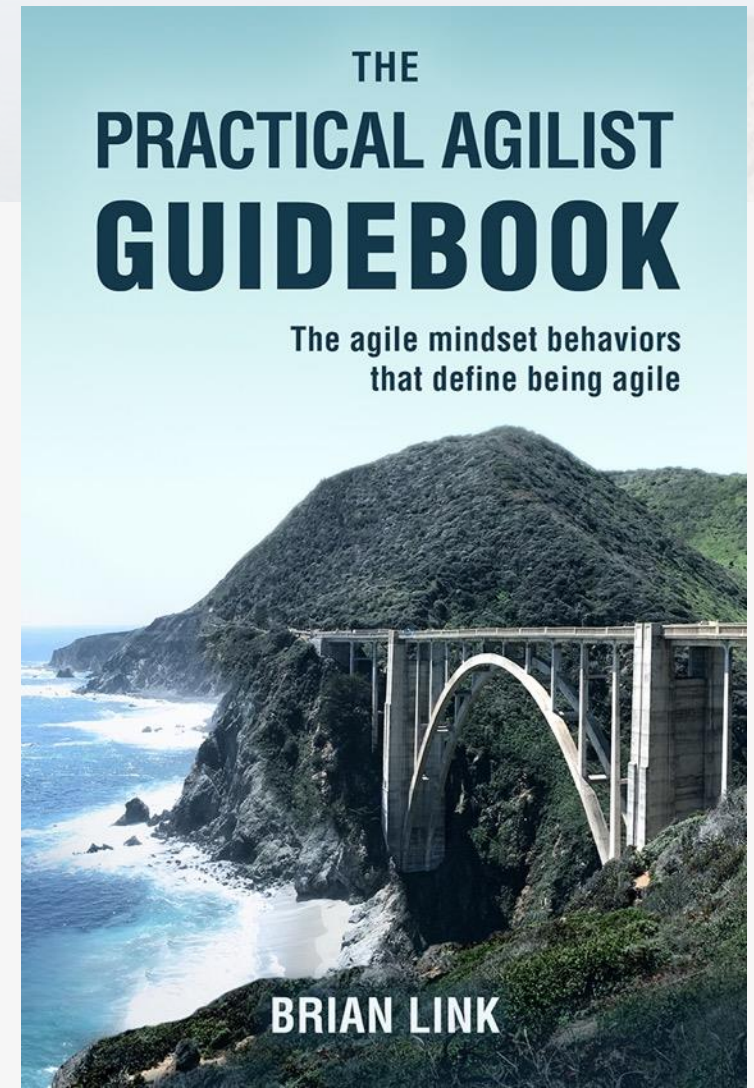
- Agile Manifesto
- Lean / Scrum / Kanban
- Extreme Programming
- Modern Agile
- Design Thinking
- DevOps
- Systems Thinking



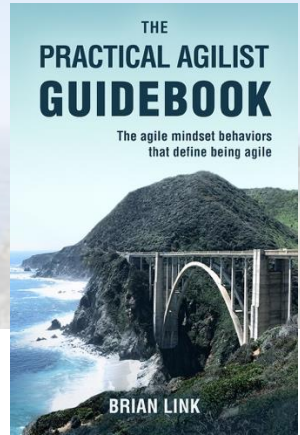
It's all about the behaviors...

You cannot BE agile...

- ...because you have a standup for 15 mins
- ...by following a framework to the letter
- ...just by declaring there is psychological safety
- ...if you don't talk to actual customers
- ...if you are afraid to fail running an experiment
- ...without trying to get feedback sooner
- ...if you don't admit you can still be better



How the book happened



- It all started with Capital One and Silicon Valley Bank...
- What to write? Start with a better agile team maturity assessment!
 - No scores! No frameworks! More for continuous improvement, learning!
 - A retro you use every 3 or 6 months to see where to focus learning
- Now what to write a *book* about?
 - Whoa. These are the same behaviors I talk to teams about as a coach!
- The same 24 agile topics for teams... become chapters!
 - Examples of low, medium, high levels of maturity for those behaviors
 - Why the topic matters, tips for leaders, tips for scrum masters, AI prompts

What do you want to learn today?

- Team Culture
 - Long-lasting teams, collaboration, psychological safety, happiness, dysfunction, and learning
- Agile Process Basics
 - Process ownership, visualization, estimation, iterative thinking, flow, continuous improvement, and experimentation
- Product Management
 - Customer-centricity, customer feedback, stakeholders, vision, strategy, value-driven, product alignment, and desired outcomes
- Value Delivery
 - Frequent value delivery, failure recovery, quality work habits, work product promotion + management, automation, governance, tech skill sharing

Choose your own AGILE adventure!



So how will this work?

- TypeForm – pick the topics that sound interesting
 - It'll rank them and I'll go through the top handful of topics
- Each topic has 2 slides
 - A quick overview of the 4 behaviors mature teams exhibit
 - Some sample images or references to third party sources for more info
 - Q&A as you like then we move on
- This is an experiment, thanks for playing!



Why Long-lasting Product Teams Matter

- Know your purpose
 - Why do you exist? What is success?
- Dedicated team members
 - All of the team is committed
- Team skillset completeness
 - All skills necessary are present
- Continuity of team members
 - No changes to the team recently

What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Know your purpose	The team works on tasks without understanding the broader purpose or strategy.	Some of the team knows the strategy and they work and how to describe it.	Everyone on the team can thoroughly explain why the team exists and specifically what success looks like for their customers and the business.
Dedicated team members	Some team members still do work for other teams, prior owners, or previous managers.	Some team members occasionally do work that is not on the team's backlog.	All team members are 100% dedicated to the team's backlog of work.
Team's skillset completeness	The team is missing key skills, roles, or it generally only relies on an outside dependency to deliver value to customers.	The team is missing one or more skills they only rely on an outside dependency to deliver value to customers.	The team has every skill they require and team to perform the end-to-end work of delivering value to its customers.
Continuity of team members	The team has had regular ongoing, or recent changes to the team's composition.	The team has had at least one member of rotating without any team member changes.	The team has been together for a year or longer without any changes, losses, or additions to the team make-up.

Team Culture

Agile Process Basics

Product Management

Value Delivery

Why Long-lasting
Product Teams
Matter

Collaboration
and Team
Independence

Psychological
Safety Inside
Teams

Psychological
Safety With
Leaders and
SMEs

Team Happiness
and Dysfunctions

A Culture of
Learning on
Teams

Process
Autonomy and
Mastery

Visualization and
Transparency

Estimation and
Expectation
Setting

Iterative Thinking
and Breaking
Work Down

Optimizing the
Flow of Work

Continuous
Improvement and
Experimentation

Customer
Centricity and
Intimacy

Communication
Channels and
Customer
Feedback

Feedback Loops
with Stakeholders

Vision, Strategy,
and Being Value
Driven

Prioritization,
Outcomes, and
Reducing
Complexity

Alignment to
Strategy and
Measuring
Progress

Repeatable,
Consistent,
Frequent Value
Delivery

Deploying Value
and Recovery
From Failure

Quality
Embedded in
Work Habits

Work Product
Organization,
Promotion, and
Management

Automated
Development
Processes and
Governance

Team Process for
Skill Sharing

Why Long-lasting Product Teams Matter

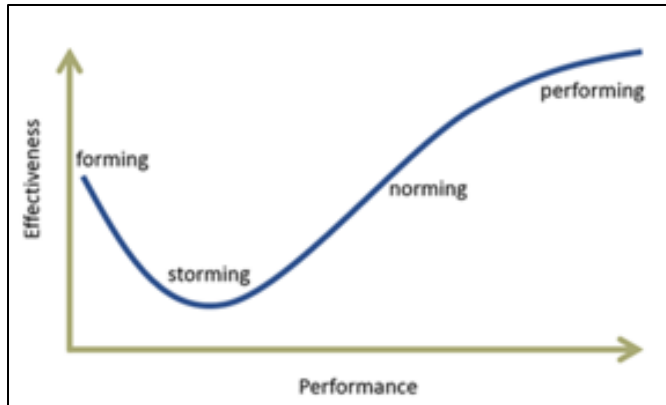
- Know your purpose
 - Why do you exist? What is success?
- Dedicated team members
 - All of the team is committed
- Team skillset completeness
 - All skills necessary are present
- Continuity of team members
 - No changes to the team recently

What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Know your purpose	The team works on tasks without understanding the broader purpose or strategy.	Some of the team knows the strategy, why they exist, and how to describe it.	Everyone on the team can thoroughly explain why the team exists and specifically what success looks like for their customers and the business.
Dedicated team members	Some team members still do work for other teams, prior projects, or previous managers.	Some team members occasionally do work that is not on the team's backlog.	All team members are 100% dedicated to the team's backlog of work.
Team's skillset completeness	The team is missing key skillsets, roles, or is potentially relying heavily on off-team people.	The team is missing one or more skills they rely on as outside dependencies to deliver value to customers.	The team has every skill they require on-team to perform the end-to-end work of delivering value to its customers.
Continuity of team members	The team has had regular, ongoing, or recent changes to the team's composition.	The team had at least six months of stability without any team member changes.	The team been together for a year or longer without any changes, losses, or additions to the team make-up.

Why Long-lasting Product Teams Matter

<http://agilist.guide/#1>



• References



Long-Lived Teams — RGalen Consulting
rgalen.com

Tuckman's stages of group development

From Wikipedia, the free encyclopedia

The forming–storming–norming–performing model of group development was first proposed by Bruce Tuckman in 1965,^[1] who said that these phases are all necessary and inevitable in order for a team to grow, face up to challenges, tackle problems, find solutions, plan work, and deliver results. Tuckman suggested that these inevitable phases were critical to team growth and development: he hypothesized that along with these factors, interpersonal relationships and task activity would enhance the four-stage model that is needed to successfully navigate and create an effective group function.^[2]

Group development

Forming

The team meets and learns about the opportunities and challenges, and then agrees on goals and begins to tackle the tasks. Team members tend to behave quite independently. They may be motivated but are usually relatively uninformed of the issues and objectives of the team. Team members are usually on their best behavior but very focused on themselves. Mature team members begin to model appropriate behavior even at this early phase. The meeting environment also plays an important role to model the initial behavior of each individual. The major task functions also concern orientation. Members attempt to become oriented to the tasks as well as to one another. This is also the stage in which group members test boundaries, create ground rules, and define organizational standards.^[2] Discussion centres on defining the scope of the task, how to approach it, and similar concerns. To grow from this stage to the next, each member must relinquish the comfort of non-threatening topics and risk the possibility of conflict.

Tuckman's stages of group development
en.wikipedia.org



From Chaos to Successful Distributed Agile Teams



Johanna Rothman
Mark Kilby

★★★★★ 26

From Chaos to Successful Distributed Agile Teams: Collaborate to Deliver
amzn.to

Collaboration and Team Independence

- Team interactivity
 - No silos, work together, pair
- Product decision making
 - Own strategic decisions
- Flow of work
 - Pull work, not assigned, own changes
- Process decision making
 - Manage own volume, how to operate

What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Team interactivity	Individuals work in silos independently without much collaboration.	The team members only occasionally discuss how they might work together or pair on work together.	The team collaborates by default, looking for ways to help each other regularly to get things done sooner.
Product decision making	Decisions are made for team members often by managers, leads, or senior team members.	The team has input in decisions that guide the work and strategic direction.	The team fully owns their strategic product decision making.
Flow of work	Work is assigned to team members, and they rarely choose their own work.	Team members pull work from the backlog. Sometimes work is assigned to team members. Key decisions are often made by leaders, the PO or SM.	Team members pull the work from the backlog and assign it to themselves. They decide to split cards on their own, actively document key decisions, and overcommunicate changes to the team.
Process decision making	The work, the flow, and volume of work is defined by someone outside the team.	The team, SM, and PO make some decisions about process. The team manages their own work, and their own work volume.	The team collectively makes nearly all decisions about the work and operates with a 'declare intent then go' manner with leadership.

Collaboration and Team Independence

<http://agilist.guide/#2>

*“Don’t move information to authority,
move authority to the information.”*

By L. David Marquet in “Turn the Ship Around!”

- References



Psychological Safety Inside Teams

- Team member trust
 - Open sharing, honest feedback, culture
- Team member contributions
 - Everyone speaks, contributes at events
- Perspective on failure
 - No fear, experiments and learning
- Conflict handling
 - Resolves peacefully, effectively

What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Team member trust	Team members fear sharing, voicing their opinion, or any kind of feedback.	Most team members trust one another and freely share feedback.	The team has strong relationships between every pair of team members and feedback is a core part of the team culture.
Team member contributions	Some members do not interact or contribute during agile events.	Most members regularly contribute during agile events.	All members consistently contribute to every agile event.
Perspective on failure	The team fears failure and anything that may be perceived as failure. As a result, they avoid experimentation.	The team sometimes sees failed experiments as learning opportunities to use to their advantage.	The team embraces a culture of experimentation and seeks feedback through early failures to build better products faster.
Conflict handling	The team avoids conflict and requires external intervention to resolve issues.	The team can handle some internal conflicts with minimal outside help.	The team resolves all internal conflicts peacefully and effectively.

Psychological Safety Inside Teams

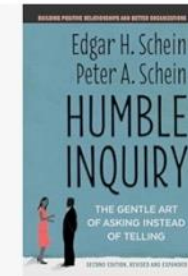


When people feel **safe** enough to raise their hands and say, “I made a **mistake**” or “I need some help.” The leader has created an **environment** where their people feel safe to be themselves.

Simon Sinek



Ideo, an innovative design company - 60 Minutes Jan. 06, 2013
youtube.com



★★★★★ 997

Humble Inquiry, Second Edition: The Gentle Art of Asking Instead of Telling (The Humble Leadership Series)
amzn.to

<http://agilist.guide/#3>

• References



How do you build psychological safety?
Building a psychologically safe workplace | Amy Edmondson | TEDxHGSE
youtube.com



Published 2016
What Google Learned From Its Quest to Build the Perfect Team (Published 2016)
nytimes.com



The power of vulnerability | Brené Brown | TED
youtube.com

Psychological Safety With Leaders and SMEs

- Leadership interactions
 - Mentors, experts, SMEs, collaboration
- Work transparency
 - Makes work visible, multiple radiators
- Manager support and buy-in
 - Supported, trusted, amplify successes
- Cross-team interactions
 - Collaborate, regular communication

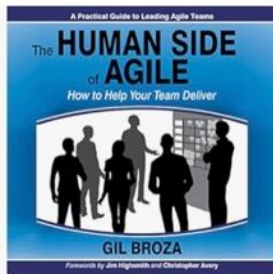
What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Leadership interactions	Team members avoid interactions with leadership.	Team members use leadership as mentors and seek feedback occasionally when needed.	The team uses subject matter experts and leaders as mentors and utilize their feedback regularly to discuss approaches and solutions.
Work transparency	The team avoids making the work visible due to fear of criticism.	The team is transparent and makes their work visible. They understand and appreciate the benefits of doing so.	The team has built multiple information radiators to share specific info with different audiences beyond the active work.
Manager support and buy-in	Managers do not actively support agile practices and either impose or fall back on traditional methods.	Managers support agile practices and engage with the team appropriately through 1:1s, sprint reviews, asking permission before attending refinements or standups.	Managers fully trust the team and communicate successes upwards and coach them through failures.
Cross-team interactions	The team avoids work from dependent teams and focuses on their own interests competitively.	The team collaborates with other teams as needed and occasionally initiates collaborative conversations in advance with other teams.	The team has regular touch points where needed to communicate thoroughly with stakeholders and dependent teams and prioritize collectively for the greater good of the company.

Psychological Safety With Leaders and SMEs

<http://agilist.guide/#4>

- Leaders earn trust through their actions, not by commanding it due to their positional authority.
- How do you lead? Do you ask questions? Do you seek to understand first or to speak and provide solutions?
- References



★★★★★ 60

The Human Side of Agile: How to Help Your Team Deliver
amzn.to

Daniel H. Pink

author of the New York Times bestseller
A Whole New Mind

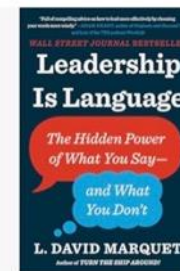
UNABRIDGED
READ BY THE AUTHOR
Hendrik

DRIVE

The Surprising Truth
About What Motivates Us

★★★★★ 8,708

Drive: The Surprising Truth About What Motivates Us
amzn.to



★★★★★ 1,182

Leadership Is Language: The Hidden Power of What You Say--and What You Don't
amzn.to

Team Happiness and Dysfunctions

- Job satisfaction
 - Love their work, job, and company
- Team dysfunction/satisfaction
 - Good relationships, even friends
- Process satisfaction
 - Invested in how they work and enjoy it
- Agile way of working satisfaction
 - Embrace whys behind agile culture

What are the maturity levels for the behaviors on your team?

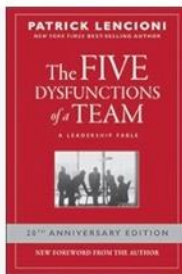
	Low	Medium	High
Job satisfaction	Team members do not like their jobs and often complain.	Team members generally like their job and the company.	Team members view it as their dream job and love where they work.
Team dysfunction/satisfaction	Team members either do not get along with each other or rarely interact.	Team members mostly get along with the rest of the team and enjoy working together.	Team members are close or are friends and often interact socially.
Process satisfaction	Team members dislike the work they do and complain about the process.	Team members have embraced how they are working and enjoy talking about it.	Team members are invested in how they work and suggest improvements.
Agile way of working satisfaction	Team members disengage from agile events and consider leaving.	Team members are engaged, involved in agile events, and are happy.	Team members think this is the best team ever and loves the way their team works together.

Team Happiness and Dysfunctions

<http://agilist.guide/#5>

“Culture is the sum total of the conversations between our employees, nothing more.”

- References



★★★★★ 13,573

The Five Dysfunctions of a Team: A Leadership Fable, 20th Anniversary Edition
amzn.to



You Are The Culture
medium.com



★★★★★ 105

Zombie Scrum Survival Guide (The Professional Scrum Series)
amzn.to

A Culture of Learning on Teams

- Managers support learning
 - Dedicated time, space for learning
- Team members actively learning
 - Learning is part of how the team works
- Opportunities to upskill
 - Many ways to learn, improve skills
- General skill sharing
 - Regular pairing, active learning backlog

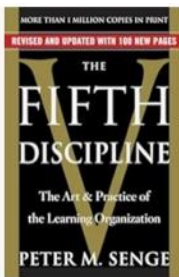
What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Managers support learning	Managers see any time not spent working as a waste and should be minimized.	Managers allow some time to be spent on learning and improving.	Managers create space for learning and help team members create learning goals that align with an individual's career.
Team members actively learning	Team members rarely work to improve skills during work hours.	Team members occasionally engage in self-improvement and learning activities.	Team members routinely engage with learning opportunities, working towards specific learning goals.
Opportunities to upskill	Training is formal, rare, and requires approval.	The company has multiple ways for teams to engage and learn, through agile communities of practice and online courses.	The company provides extensive career path planning and learning resources that target individual roles and a large variety of skills.
General skill sharing	Team members do not share skills with other team members.	Team members occasionally pair or find ways to share skills with each other.	Teams have a learning backlog, pair regularly, and often dedicate time to skill sharing.

A Culture of Learning on Teams

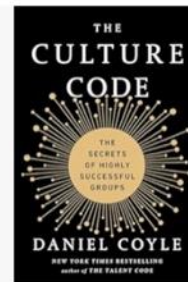
<http://agilist.guide/#6>

- The CIO of M&T Bank in Buffalo, Mike Wisler, once said, “We are a learning organization that happens to be a bank.” This statement alone, communicated to hundreds of middle managers, set the stage for it to be OK for every team to focus on continuous improvement, training, and investing time to build their skills.
- References



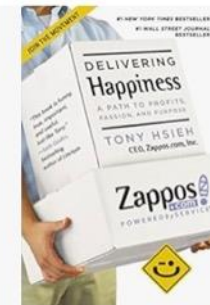
★★★★★ 2,353

The Fifth Discipline: The Art & Practice of The Learning Organization
amzn.to



★★★★★ 7,796

The Culture Code: The Secrets of Highly Successful Groups
amzn.to



★★★★★ 5,862

Delivering Happiness
amzn.to

Process Autonomy and Mastery

- Embracing agile events
 - All team participates, shares facilitating
- Ownership of agile
 - All grasp the spirit of agile, adapt roles
- Team mindset
 - Works together to plan, refine, etc.
- Process improvement
 - Embraces customizing agile to suit needs

What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Embracing agile events	The team skips some events due to a perceived lack of time or a lack of engagement.	The team participates in all agile events and is generally engaged.	The team customizes agile events to fit their needs and is comfortable rotating facilitators.
Ownership of agile	The team is largely mechanical and follows events and processes rigidly, without understanding the purpose.	The team understands the purpose of events and uses agile roles effectively.	All team members fully grasp the spirit of the agile process and adapt agile roles as needed.
Team mindset	Planning and refinement are infrequent and often done only by the Product Owner. Team members attend sporadically and don't contribute much.	The whole team is involved in planning and refinement meetings, and input is given freely and valued.	Planning and refinement happen fluidly on demand, or just-in-time, and team members are skilled in breaking down stories.
Process improvement	The team rarely inspects or improves their process, and changes are infrequent.	The team occasionally identifies and makes changes to improve their process.	The team is comfortable adapting and customizing their process regularly to suit their needs as they evolve.

Process Autonomy and Mastery

守

Learn



破

Detach



離

Transcend



<http://agilist.guide/#7>

守

Learn



破

Detach



離

Transcend



• References



Visualization and Transparency

- Use of information radiators
 - Current work visible, info for audiences
- Working agreements
 - Living document, updated regularly
- Document/communicate decisions
 - Share important info, auditable
- User story writing
 - Small, outcomes, AC's, source of truth

What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Use of information radiators	The team does not make the work visible, or it is not up to date.	There is one central, up-to-date source of the team's work.	The team uses multiple information radiators to share different levels of information with different audiences.
Working agreements	There is no formal, visible working agreement.	The team built a working agreement together and made it visible.	The team reviews and updates their working agreement every 3 months and anytime a team member <u>changes</u> .
Documenting and communicating decisions	Business or technology decisions or are not documented or shared.	Important business or technical decisions are documented in a common team space.	Documented decisions are communicated well, organized, and auditable, linked to the work items where they originated.
User story writing	User stories read more like requirements and lack clarity, outcomes, or acceptance criteria.	User stories are clear, express desired outcomes, and use acceptance criteria whenever useful. Stories are usually small, less than a week of work.	User stories are small, have clear outcomes and use acceptance criteria almost always. Stories are the one source of truth for the delivery history, decisions, and auditable history.

Visualization and Transparency

Three-C's Revisited
© Mar 26, 2019 • [Practices]

My classic article on *Card*, *Conversation*, *Confirmation* reflects the way we handled stories in Extreme Programming. We wrote a quick description of the feature on a card, talked it through with our Customer (think Scrum Product Owner), then created concrete testable examples to confirm that the feature did what we and the Customer wanted. For us, the Card came first.

Today, as the posting from the Organization That Should Know Better indicates, people fall more and more into writing requirements documents on the Card. The idea, whether expressed out loud or not, is that the story card should contain enough detail that it "makes sense" to use a standard format. C3 tried a standard format card in something like 1998 and realized it was a bad idea then, and we said so.

Chet and I were talking today about the Organization That Should Know Better's tweet, and about how one might do stories today, especially in

Three-C's Revisited
ronjeffries.com

chethendrickson
@chethendrickson

Replying to @wouterla @sebrose and 6 others


@KentBeck said imagine we have built the best payroll system ever, and we go talk to the users. They would tell us stories of how they used it. We would write them down and then go back in time and build that.

6:06 PM · Oct 24, 2019 · Twitter for iPhone

<http://agilist.guide/#8>

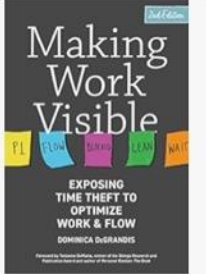


• References



★★★★★ 679

Extreme Programming Explained: Embrace Change, 2nd Edition (The XP Series)
amzn.to



★★★★★ 428

Making Work Visible: Exposing Time Theft to Optimize Work & Flow
amzn.to

- **I** – Independent
- **N** – Negotiable
- **V** – Valuable
- **E** – Estimable
- **S** – Small
- **T** – Testable

Estimation and Expectation Setting

- Completing work
 - Flow is predictable, sets expectations
- Story points or #noestimates
 - Consistent relative sizing or count cards
- Expectation setting and ownership
 - Metrics, roadmaps set short/longterm
- Estimation mindset
 - Directionally correct, usefully wrong

What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Completing work	The team does not deliver work reliably and often fails to set expectations about when work will be done.	The team delivers completed work and sets expectations reliably most of the time with some exceptions.	Completed work flows regularly and predictably. Expectations are reliably given internally and externally with very few exceptions.
Story points or #noestimates	Story points are tied to hours or not used properly to set expectations and maintain a sustainable pace for the team.	Story points are relatively sized, rough estimates, help gain a shared understanding of the work, set expectations, and maintain a steady pace.	Story points are not used because the team has perfected the art of collaborating and slicing stories to a size that counting cards suffice.
Expectation setting and ownership	Estimates are either dictated by someone outside the team or done by a small subset of the team.	The team owns short term expectations using metrics. Long term goals are either dictated to them or co-owned with leaders.	The team owns both short- and long-term expectation setting <u>through the use of</u> metrics, release plans, and Now/Next/Later roadmaps.
Estimation mindset	Estimating takes a significant amount of time (i.e. an hour or more) each week.	The team minimizes the time spent estimating and realizes being directionally correct and usefully wrong is enough.	The team spends no time on the process of estimating.

Estimation and Expectation Setting

<http://agilist.guide/#9>



- References



What are the Three Amigos in Agile?
agilealliance.org



Why use story points? Because everyone
sucks at estimating!
medium.com

Iterative Thinking and Breaking Work Down

- Consistency of size
 - Work is always small, done in a few days
- Vertical slicing ability
 - Focus on incremental value, hypotheses
- Work with PO
 - Understands purpose, gets direct feedback
- Embracing feedback
 - Proactively seeks feedback for decisions

What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Consistency of size	User stories are sometimes large enough to linger in the active work of the team for more than 2 weeks.	User stories are almost always finished inside of 2 weeks and the team has established a flow of getting work done regularly and seeking feedback.	User stories are as small as they can be to still deliver value of some kind and generate feedback, most often taking less than 3 days to complete.
Vertical slicing ability	Work takes the form of tasks or large chunks of work more than small iterative pieces. Delivery and quality portions of work are sometimes still separate user stories.	Work most often satisfies the typical INVEST standard and focuses on a desired outcome with an iterative approach to producing customer value incrementally.	User stories use a vertical slicing approach as needed to deliver small portions of value or, when useful, more in the form of a hypothesis than requirements to determine the best approach.
Work with PO	The team rarely asks the Product Owner for feedback on partially completed work or about stories before marking them as done.	The team sometimes communicates with the Product Owner to better understand the desired outcome for work while in progress, and usually as work is completed to ensure it satisfied the purpose of the story.	The team has daily conversations with the Product Owner as needed to show incomplete work, get feedback, and ask clarifying questions. The team always confirms a story meets the desired outcome before it's done.
Embracing Feedback	The team is reactive and completes work as instructed without asking about the why behind the work or seeking to know what they should be trying to learn.	The team only sometimes seeks feedback to decide what to do more of and, importantly, what not to do. The team only occasionally changes their mind about upcoming work based on feedback.	The team proactively seeks to understand what they need to learn so they are better prepared to make decisions about what to do next. They use feedback from completed work to change their minds often about what to do next.

Iterative Thinking and Breaking Work Down

<http://agilist.guide/#10>

HOW TO SPLIT A USER STORY

1. PREPARE THE INPUT STORY

- Start by writing down the user story in a clear and concise way.
- Consider the user's perspective and the value they are seeking.
- Use the "As a [user role], I want to [action] so that [benefit]" format.

2. APPLY THE SPLITTING PATTERNS

- NORMALIZE STEPS:** Can you split the story into smaller, sequential steps?
- USER PERFORMANCE:** Does the user have a specific performance goal?
- OPERATIONS:** Can you split the story into smaller, sequential operations?
- BUSINESS RULE VARIATIONS:** Can you split the story into smaller, sequential business rules?
- VARIATIONS OF DATA:** Can you split the story into smaller, sequential data variations?
- BREAK UP & SPLIT:** Can you split the story into smaller, sequential parts?

3. EVALUATE THE SPLIT

- Is the split user story clear and concise?
- Is the split user story achievable?
- Is the split user story valuable?

humanizing **WORK**
www.humanizingwork.com

The Humanizing Work Guide to Splitting User Stories
humanizingwork.com

- References

FIFTY QUICK IDEAS TO IMPROVE YOUR USER STORIES

★★★★★ 359

Fifty Quick Ideas to Improve Your User Stories
amzn.to

Why Split Big User Stories?
medium.com

Optimizing the Flow of Work

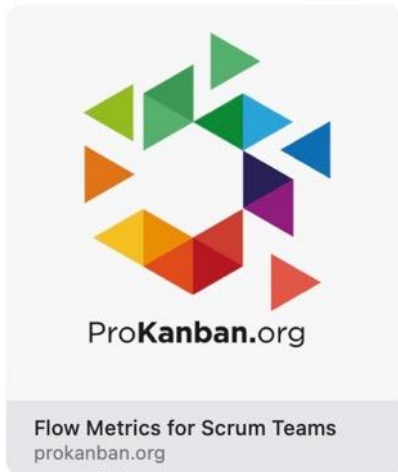
- Definition of done
 - Used always and revisited frequently
- Definition of ready
 - Becomes unnecessary with great habits
- Work in progress limits
 - Team uses as tool to get work done sooner
- Utility of metrics
 - Adjusts process by being data-driven

What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Definition of done	The team either does not use a Definition of Done, it is not very thorough, hasn't been reviewed recently, or it is not followed consistently.	The team has a Definition of Done and uses it regularly to make sure almost all work satisfies the DoD before being marked as done.	The team revisits their Definition of Done at least quarterly and adjusts as they discover new constraints or challenges that need to be incorporated.
Definition of ready	The team either does not have a Definition of Ready, or if they do have one, it may be used as an excuse to not start work or is not used consistently.	The team has a healthy practice of ensuring work is ready and important, well-defined, and well-understood before it's added to the team's list of ready work.	Everyone on the team knows why the current work was deemed ready. The PO controls the readiness of the flow of work. The formality of a DoR is no longer necessary because the team has regular routines that flow the work.
Work in progress limits	The team does not monitor the amount of work in progress or use it consistently to change the behavior of how they work.	The team has an understanding and appreciation for monitoring their WIP limits and sometimes hold each other accountable to completing work before starting new work.	The team adheres to WIP limits consistently, monitors, and adjusts them as necessary. They work together as a team to get stories done sooner before starting new work.
Utility of metrics	If the team is tracking story points, velocity, cycle time, or lead time, they are not being used or measured consistently in a way that helps the team make changes to the way they work.	The team is consistently completing work according to stabilizing velocity or cycle time measures. The team captures a trend of these metrics over time to help them adjust their process and workload.	The team brings cumulative flow diagrams or multiple metrics into their retros to make sure they are delivering value quickly and customizing their process as needed.

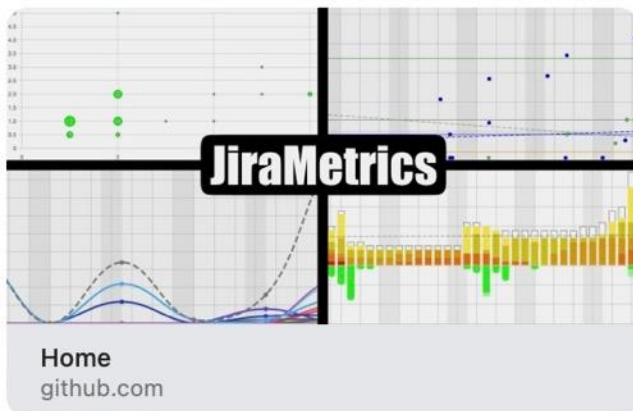
Optimizing the Flow of Work

<http://agilist.guide/#11>



- **WIP:** The number of PBIs started but not finished.
- **Cycle Time:** The amount of elapsed time between when a work item started and when a work item finished.
- **Work Item Age:** The amount of elapsed time between when a work item started and the current time.
- **Throughput:** The number of PBIs finished per unit of time. Note the measurement of throughput is the exact count of PBIs.

References



POTENTIALLY SHIPPABLE

- + TEST
- + STATIC ANALYSIS
- + INTEGRATION
- + PACKAGING
- + STAGING
- + CUSTOMER DOCUMENTATION
- + RISK EVALUATION
- + CODE
- + APPROVAL
- + UPDATE MARKETING MATERIAL
- + REGULATION
- + PREPARE FOR CUSTOMER FEEDBACK

UNIT INTEGRATION
USER ACCEPTANCE
ACCEPTANCE
SYSTEM
PERFORMANCE
STABILITY
USABILITY
STRESS
MONKEY
SMOKE

UNDERLINED ONES ARE THE DEFINITION OF DONE

Definition of Done
less.works

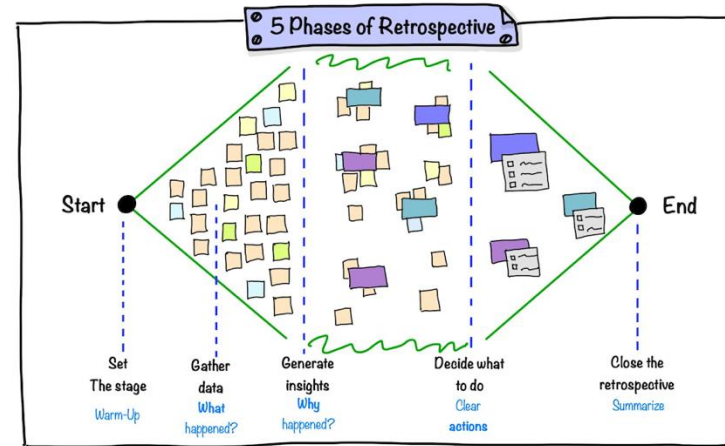
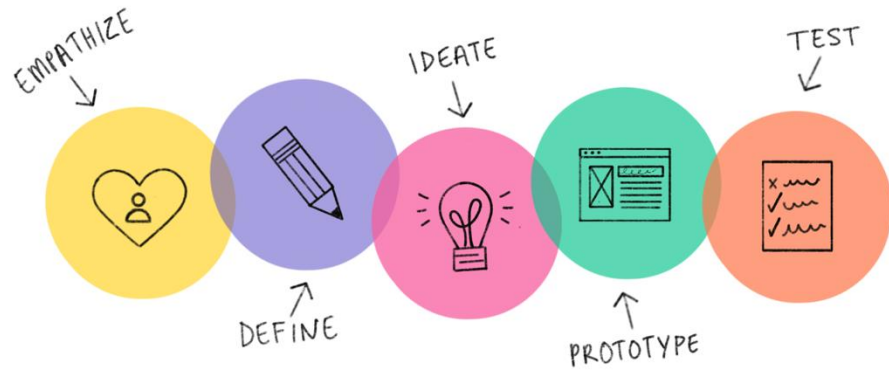
Continuous Improvement and Experimentation

- Use of retros
 - True inspect+adapt, in real time when needed
- Design thinking
 - Con't discovery, experiments, hypotheses
- Use of failure
 - Learning! Not afraid to learn something sooner
- Data-driven
 - Evidence based decisions wherever possible

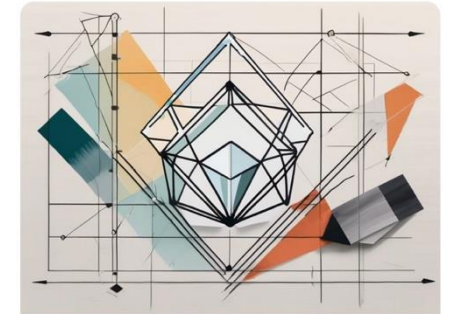
What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Use of retros	The team skips retros and/or avoids the topic of improving their own process.	The team uses the retrospective event to inspect themselves, their process, and has an open conversation about what changes, if any, they should make to work better together.	The team addresses important process changes in real time as needed. Retros are used to discuss less urgent process changes, team building, experiments, or other points of improvement.
Design thinking	There is little to no focus on the concepts of Design Thinking (empathize, define, ideate, prototype, test) to design and test various concepts of future work with customers in mind.	Design Thinking is a part of how the team works. The team tests hypotheses with experiments designed to validate potential customer outcomes and does so before developing new features.	The team utilizes a dual-track agile approach or continuous discovery and continuous delivery to integrate both short- and long-term experiments into the backlog prioritization process.
Use of failure	The team (and often their managers) have strong opinions about "failure" and strive to avoid anything that looks or feels like failure.	The team is learning that failure means there is something to learn. They adapt and react quickly to failure without much stigma or fear of repercussion.	The team purposefully writes stories in a way that looks to learn sooner so they can build more of what matters to customers and less of what does not. They are not afraid to share when experiments were not successful as intended.
Data-driven	There is a lack of any data-driven decision making. Decisions to change direction or validate whether the product is being used as intended or not are rarely provable with data.	Decisions about the product are sometimes made because of evidence produced through data. A data-driven mentality is starting to become part of the product prioritization process.	It is rare that any features are built based on a guess or an instinct of what a customer might like. Instead, the team has made it part of their practice to create an evidence-based, data-driven approach to almost all their big product decisions.

Continuous Improvement and Experimentation



<http://agilist.guide/#12>



The Triple Diamond Design Process: A Comprehensive Guide
launchnotes.com

• References



IDEO Design Thinking
designthinking.ideo.com



Agile Retrospectives: The Heart of Team Improvement
youtube.com



Agile Retrospectives, Second Edition: A Practical Guide for Catalyzing Team Learning and Improvement
amzn.to



This Is Service Design Doing: Applying Service Design Thinking in the Real World
amzn.to

Customer Centricity and Intimacy

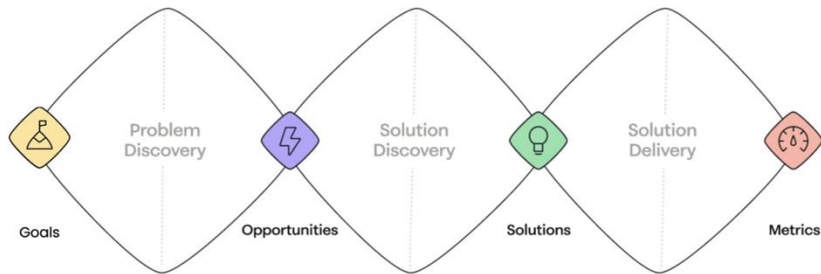
- Problem solving approach
 - Customer experience over company problems
- Customer personas
 - Relatable customers integrated with process
- Customer outcome focused user stories
 - Stories target persona outcomes and verified
- Customer research
 - Connect with customers, understand journey

What are the maturity levels for the behaviors on your team?

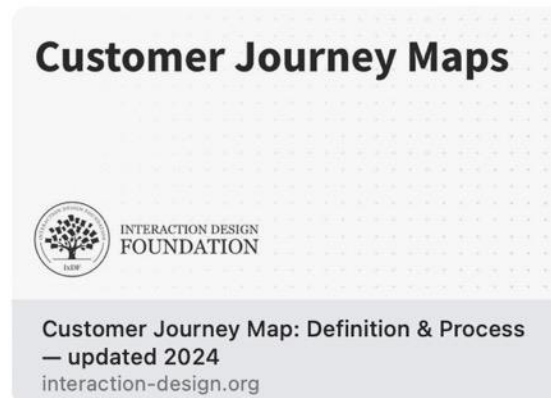
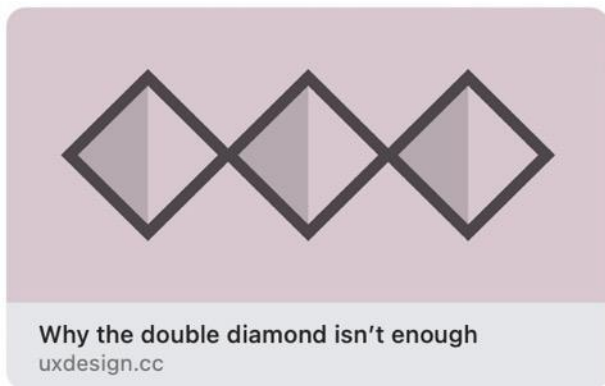
	Low	Medium	High
Problem solving approach	Team follows a traditional problem-solving approach, identifying objectives from a company centric perspective first. Then identifies concepts and fits solutions to the customers.	Team is starting to incorporate learnings from the customer to decide which of their problems to address. The customer experience has started to influence the product direction.	The team uses a process to understand the customer experience first and lets that drive how they can better serve the customer's journey.
Customer personas	There is little to no knowledge on the team about who the customer is. Some team members perform work perfunctorily as if it were just a list of tasks without knowing who benefits.	The team has either created customer personas or somehow have dedicated some time to understanding more thoroughly who their customers are and what motivates them.	Every team member is aware of the distinct personas and customer types that may use the product. These personas are updated as necessary and included in regular team conversations involving user stories and prioritization.
Customer outcome focused user stories	User Stories are written without firsthand knowledge of the customer or without anticipation about how the work will be used.	User Stories are in part crafted based off on customer feedback. There is a customer feedback loop that directly impacts the direction of the product.	Some user stories, designed for targeted type(s) of customers, are written to focus on specific outcomes and the team follows up with the customer type(s) to confirm or deny the impact.
Customer research	If there is any inspiration about the customer in terms of who they are and what they want, it is guesswork or just always the way things have been done.	There has been some level of research done with a sample of customers to understand their experiences in context of the team's work, know who they are and what they want.	There is a thorough and regularly updated set of documentation about who the customer is, based on directed research, regular interactions, and customer experience journey maps.

Customer Centricity and Intimacy

<http://agilist.guide/#13>



• References



Communication Channels + Customer Feedback

- Feedback loop with customers
 - Regular communication, multiple channels
- Customers integrated with the process
 - Show WIP get early feedback, beta releases
- Customer list
 - Actual customers, peer-to-peer community
- Level of intimacy
 - 1:1 convos, observe in action experiences

What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Feedback loop with customers	There is no communication channel between the customer and the team.	There is an established communication channel between the customer and the team, either a regular real-time dialog or through asynchronous messaging that customers can and do directly provide feedback to the team.	The team has established regular communication channels with customers through multiple mediums to make it easier for two-way communication.
Customers integrated with the process	The team does not inform customers about changes or upcoming releases. No customers have been asked directly for feedback on any work the team has done.	The team provides updates to the customer to inform them of changes. Customers have the opportunity to provide feedback to the team to request features and/or discuss plans for the product.	The team is in a regular routine of showing customers work in progress to elicit feedback before features are complete and before the value is produced. Early releases or betas may even be made available to a select list of customers.
Customer list	The product owner or team acts as proxy on behalf of the customer, making assumptions.	There is at least one named customer the team has contacted and established as a proxy to help make decisions on a day-to-day basis.	A private community of customers has been established. They are invited to real-time / roundtable conversations to hear product roadmap updates and provide feedback on long-term strategic plans for the product.
Level of intimacy	There are no established relationships with customers to use for early feedback or research.	The team has conducted surveys of customers and had multiple 1:1 conversations or interviews to better understand their perspective and context.	Team members have been invited and visited the customer's workplace to observe the environment, the behaviors, and see their problems first-hand. Could also involve capturing phone call audio with permission or other immersive experience to learn more about their customer's context.

Communication Channels + Customer Feedback

<http://agilist.guide/#14>

Online Shopping Customer Journey Map

Stages of Journey	Motivation	Searches for Websites	Browses the Site	Evaluates Products	Pay	Receives Products				
Activities	Wants to buy a Christmas gift for a friend	Searches products keywords on search engines	Clicks the 1st ads. result.	Checks online again and go to the 1st. original result	Checks out ongoing deals and hot sale products	Opens a product page to check product details.	Opens more product pages to compare info.	Processes to pay a preferred product.	Contacts online service for help.	Product delivered.
Feelings	Very Happy	Overall Satisfied	Unhappy	Happy	Surprise	Frustrated	Pleased	Frustrated	Satisfied	Happy
Experiences	Excited to give a surprise to the friend.	Happy to see many options. Confusing about how to choose.	Annoyed at the useless info.	Happy with the informative shopping website. Not sure about how to find the best price.	Surprise to see lots of discounts	Frustrated about reviews from other customers.	Pleased with usages & designs. Sad to see out of stock products.	Frustrated to find only the Paypal payment option.	Satisfied with bank card payment. Feels unhappy to wait for a bit of long time.	The friend successfully received the gift before Christmas.
Customer Expectations	Easily to obtain discount news.	More user-friendly search engine.	Allow to hide unnecessary ads.	Clear and Innovative website design.	More discounts during holidays.	Higher quality products.	Faster supply chain system.	More payment choices	Faster website speed.	Free delivery fee in the future.

- References



SENSE & RESPOND
How Successful Organizations Listen to Customers and Create New Products Continuously
JEFF GOTHELFF & JOSH SEIDEN
★★★★★ 188

Sense & Respond: How Successful Organizations Listen to Customers and Create New Products Continuously
amzn.to

7 Ways to Analyze a Customer-Journey Map



nngroup.com
NN/g

7 Ways to Analyze a Customer-Journey Map
nngroup.com

Customer Journey Maps



INTERACTION DESIGN FOUNDATION

Customer Journey Map: Definition & Process — updated 2024
interaction-design.org

Feedback Loops with Stakeholders

- Invite to sprint reviews
 - Engaged in events, including refinement
- Regular communication
 - Cadence as needed, scheduled, ad hoc
- Feedback loop
 - Show WIP for feedback, share roadmap
- Vision sharing
 - Make aware, ask for help evangelizing

What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Invite to sprint reviews	The team is either unsure of who their stakeholders are or have not reached out to invite them to sprint reviews.	Stakeholders and/or SMEs occasionally attend Sprint Reviews or regular meetings and are invited to backlog refinement meetings as needed.	Stakeholders and SMEs find it valuable to be invited to Sprint Reviews or regular meetings and attend regularly and engage with the team during those events.
Regular communication	The team has not established any communication channels or regular interactions with stakeholders.	The team and stakeholders communicate asynchronously as needed.	The Product Owner and/or team have a regular dialog with key stakeholders often.
Feedback loop	Stakeholders are not yet regularly invited to any agile events, therefore there is no feedback from internal stakeholders.	The team asks for feedback from stakeholders and SMEs regularly on recently completed work and ideas they are considering. The feedback is considered and often helps influence the direction of the product or in prioritization.	The team asks for feedback from stakeholders and SMEs regularly on recently completed work, ideas they are considering, as well as product roadmap updates, and incomplete work still in progress. The feedback is considered and often helps influence the direction of the product or in prioritization.
Vision sharing	The team has not communicated with stakeholders to share the vision and include them in plans.	The team has communicated with stakeholders to share the vision and include them in upcoming plans as needed.	The stakeholders are fully aware of the product vision and help evangelize it to their colleagues and coworkers.

Feedback Loops with Stakeholders

<http://agilist.guide/#15>



SAYING NO TO STAKEHOLDERS

romanpichler

5 Tips for Saying No to Stakeholders
youtube.com



Product Owner + Stakeholders

How do these roles collaborate? What impact is there if they don't collaborate?

YDS: How Do a Product Owner and Stakeholders Collaborate?
youtube.com



MY AGENDA FOR SPRINT REVIEWS

Sprint Review Agenda: How to Make the Most of Your Time
youtube.com



HOW TO BUILD A STAKEHOLDER ENGAGEMENT STRATEGY

CORE STAKEHOLDERS
HIGH FREQUENCY = Every 1-2 weeks
HIGH TOUCH = 1:1s

How to Build a Stakeholder Engagement Strategy
youtube.com



SCRUM MASTERY

Geoff Watts

From Good to Great
Smart Leadership

2nd Edition

★★★★★ 320

Scrum Mastery (Geoff Watts' Agile Mastery Series)
amzn.to



Mountain Goat Software

Scrum & Agile Training | Agile Resources | About Us | Our Blog | Training Schedule | Sign In

Agile > Sprint Review Meeting

Sprint Review Meeting

Learn About Agile

An Introduction to Agile and Scrum

The Scrum Framework >

Scrum Roles: An Overview

Scrum Master Role and Responsibilities

Product Owner Role and Responsibilities

Scrum Team Role & Responsibilities

The Chicken and the Pig

Scrum Activities: An Overview >

Sprint Planning Meeting

What Happens in a Sprint Review?

The sprint review is an inspect and adapt opportunity that occurs at the end of each sprint. The purpose of a sprint review is to gather feedback on (inspect) newly added features. Review participants then discuss the implications of the features and feedback on the [product backlog](#) and project schedule, which influences what will be built next (adapt).

For example, based on feedback, a [product owner](#) may elect to release the current version rather than wait a few more sprints as originally planned. Or the product owner may decide to hold off on releasing to improve the newly added features a bit. Similarly, stakeholders may love some aspect of the recently developed features and want that carried into other parts of the product.

Sprint Review Meeting
mountaingoatsoftware.com



HOW TO BUILD A STAKEHOLDER MAP

STAKEHOLDER MAP

PRODUCT

PROJECT

STAKEHOLDERS

Product Pathways

How to Build a Quick and Powerful Stakeholder Map
youtube.com

Vision, Strategy, and Being Value Driven

- Understanding product vision
 - Everyone knows vision+purpose, it gets shared
- Awareness+alignment to big picture
 - Team has say in goals, knows how they fit, metrics
- Value driven
 - Steers priorities, is validated, confirmed w/ users
- Strategy ownership
 - Vision ownership, full PM, leadership support

What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Understanding product vision	The team is unsure of their product vision, or it is exclusively in the head of the product owner.	The team generally knows their product vision and why it matters.	Each team member understands their product vision, the business purpose, and the team has documented and shared this vision so others can see.
Awareness + alignment to big picture	The team is unaware how their work may impact the bigger picture in their part of the company. And if they think they know, there are no measures in place to prove or disprove the impact.	The backlog is aligned to the current strategic goals for their team of teams or section of the company. These strategic goals are visible, time-bound, and measurable (e.g., OKRs).	The team contributes to their team of teams or divisional strategic goals, understands how they fit into the big picture, and the team measures the impact they have against those higher goals.
Value driven	The team's product delivery is not driven by delivering value but completing tasks or requirements.	The team is starting to understand how to determine whether the work they deliver is in fact valuable and is starting to measure the value delivered.	The team theorizes what work will be valuable and uses that when prioritizing. The team determines whether the work they deliver is in fact valuable and confirms the value delivered through informal and formal means with the customer (e.g., conversations and measurable metrics)
Strategy ownership	The team tends to get strategic direction from managers as to what to work on next.	The product owner and team own their product vision and have the support of their leaders. Most product decisions are owned by the team but not all.	The product owner and team lead own the product completely, including specific research, competitive analysis, and exhibit foresight about how the product should change to meet market needs then adjust as needed.

Vision, Strategy, and Being Value Driven

<http://agilist.guide/#16>



MindSpring Presents: "Greatness" by David Marquet
youtube.com



Product Strategy is About Saying No - Des Traynor - Business of Software
youtube.com



The Relationship Between OKRs and Strategy
linkedin.com

- References



Who Does What By How Much?: A Practical Guide to Customer-Centric OKRs
amzn.to

Prioritization, Outcomes, Reducing Complexity

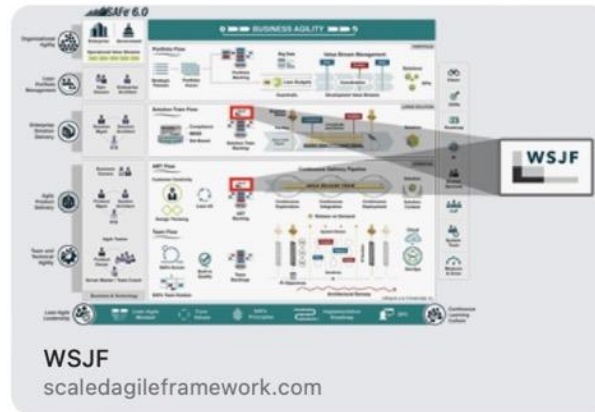
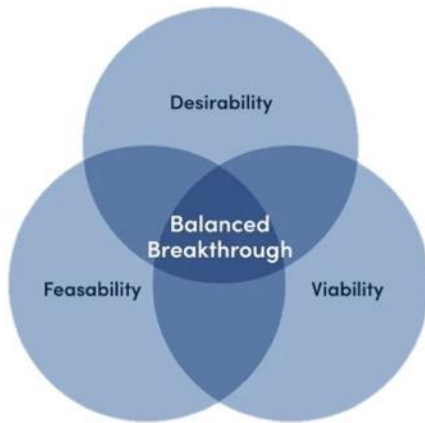
- Deliberate methods or algorithms
 - Science not gut to prioritize w/ strategy, goals, vision
- Refinement
 - Habit of breaking down, understanding, SMEs if need
- Collaboration
 - Whenever needed, Stakeholders + customers too
- Handling complexity
 - Multiple product priorities, portfolio balancing, ST/LT

What are the maturity levels for the behaviors on your team?|

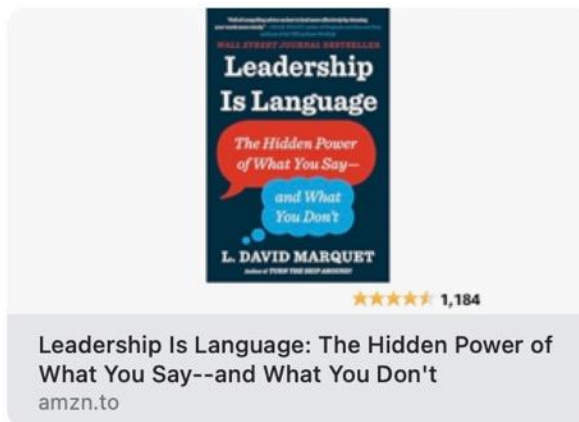
	Low	Medium	High
Deliberate methods or algorithms	The backlog does not have a system as to how it is prioritized and sequenced. It may be whimsical, mostly reactive, driven by the loudest complainers, or have no system at all.	The product owner and team discuss priorities and focus on their own perspective of product prioritization, perhaps aligned to strategic goals, using themes, incorporating quality, stability, and things the team feels are value driven.	The product owner uses a process or model such as the balanced breakthrough model (desirability, feasibility, viability) to prioritize work more scientifically in the backlog. The PO takes input from every possible source to make sure they aren't missing important work while prioritizing.
Refinement	The team does not meet regularly to refine the backlog. And if they do, some work still starts that is not thoroughly understood, has significant unknowns, or is generally too large.	The team conducts backlog refinement, reviewing priorities, breaking down work, discussing items enough to know what changes need to be made to the backlog. This process may still be time consuming and irregular.	The team meets consistently to conduct backlog refinement, reviewing priorities, breaking down work, discussing items enough to know what changes need to be made to the backlog. This process is deliberate, likely done only as required with targeted outcomes and often includes stakeholders or other SMEs as needed.
Collaboration	There is very little collaboration among team members to determine priorities for the team.	The team collaborates more than just during refinement to discuss the work, needs, and changes. The team has also started collaborating with stakeholders and/or customers to include their perspective in prioritization.	The team collaborates regularly as ideas emerge that should become stories. Stakeholders and/or customers are included in prioritization as needed. Stakeholders and SMEs have become closer with the team and are more readily available, often pulled into conversations to validate or confirm assumptions about work.
Handling Complexity	As the team supports priorities from multiple sources or has multiple products, priorities are driven by biggest fire or loudest voice.	As the team supports multiple sources of input or multiple products, there is an effort made to balance the priorities of work. There may still not be rigor or science, however.	As the team supports multiple sources of input, multiple products, or areas of complexity, the Product Owner and team have learned to treat them like a portfolio, prioritizing work inside each of the competing areas to balance the big picture with short- and long-term goals.

Prioritization, Outcomes, Reducing Complexity

<http://agilist.guide/#17>



- References



Alignment to Strategy and Measuring Progress

- Roadmaps
 - Now/Next/Later, includes measures, no dates
- Applying OKRs
 - Drive priorities, use to say no, dovetails w/higher
- Treatment of epics
 - Business value, strategic tool, Lean Portfolio
- Market + competitive analysis
 - Know alternatives, pulse of market, differentiate

What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Roadmaps	There is no product roadmap shared outside the team other than a sequencing of epics in the backlog.	The team has a product roadmap. It may simply be a sequencing of epics with rough dates and rough outcomes to help set expectations with stakeholders and/or customers.	The product roadmap follows a Now/Next/Later format (or is otherwise not tied to absolute dates and is high-level and strategic in nature) and is defined by a sequencing of product specific outcomes tied to meaningful measures to know when that portion of the roadmap has achieved its expected value through what's been delivered.
Applying OKRs	There are no strategic objectives with measurable key results to model or align to.	The team has their own OKRs that they use to set priorities and help prioritize their work.	OKRs are the primary way the team prioritizes what to work on. The team publishes their current strategy and the related OKRs, then delivers against that strategy defined by the objectives and key results and says no (or not yet) to any work that falls outside of them.
Treatment of epics	Epics are mostly used as containers for work that is not yet defined. The business value of Epics is usually not clear.	Epics have business value meaning and used as large user stories that have yet to be broken down. They often include desired outcomes, define future intentions, and can be closed when finished. Only sometimes are Epics used as evergreen containers.	In addition to defining business value, Epics are a strategic tool that enable Lean Portfolio Management (a formalized demand and intake management process), helping the company to officially fund or not fund specific pieces of work and assign those strategic pieces of work to different teams (or teams of teams).
Market + competitive analysis	If there is a Product Strategy, it is internally focused and does not yet account for market or competitive analysis.	The Product Owner and/or team has done some competitive product research and analysis to understand their customer's alternatives.	The Product Owner understands the pulse of the market, its competitive forces, and how the customers differentiate between competing products.

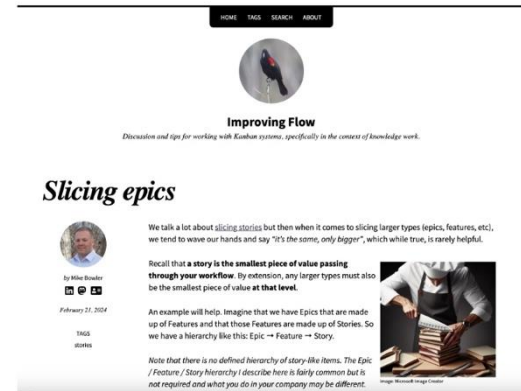
Alignment to Strategy and Measuring Progress



Why the secret to success is setting the right goals | John Doerr | TED
youtube.com



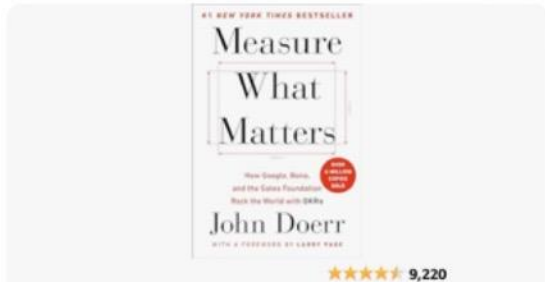
Why I Invented the Now-Next-Later Roadmap
prodpad.com



Slicing epics
improvingflow.com

<http://agilist.guide/#18>

• References



Measure What Matters: How Google, Bono, and the Gates Foundation Rock the World wi...
OKRs
amzn.to



Lean Portfolio Management
scaledagileframework.com

Repeatable, Consistent, Frequent Value Delivery

- Deployment frequency
 - Deliver value predictably, daily, < weekly
- Repeatable process
 - Automate publishing, on demand anytime
- Lead time to production
 - Predict w/in days new->upcoming release

What are the maturity levels for the behaviors on your team?


	Low	Medium	High
Deployment frequency	Deployment of value is unpredictable and not on a set schedule. May be measured in months.	Deployment of value is fairly predictable and most often four weeks or less.	Deployment of value is predictable and occurs very frequently, measured in days or multiple times per day, almost always at least once per week.
Repeatable processes	For any value produced, there is a manual process involved to publish it to end users.	The team has made it a priority to automate the delivery of value. There is a deployment process that is at least partially automated.	The deployment process is almost entirely automated with manual interventions only as desired and necessary for the process. The team can release new value on demand whenever they are ready.
Lead time to production	After work has started, there is no predictable way or reliable pattern to determine when it will be available for end users.	There is an established pattern and a repeatable process that allows the team to predict within a matter of a week or two when newly started work will be made available to its end users.	There is a pattern and a repeatable process that allows the team to predict within a matter of a few days when newly started work will be made available to its end users.

Repeatable, Consistent, Frequent Value Delivery



Get better
at getting
better


DORA Guides
dora.dev



What is DevOps?

DevOps Model Defined

What is DevOps? - DevOps Models Explained - Amazon Web Services (AWS)
aws.amazon.com



The Three Ways: The Principles Underpinning DevOps
Article by Gene Kim
itrevolution.com

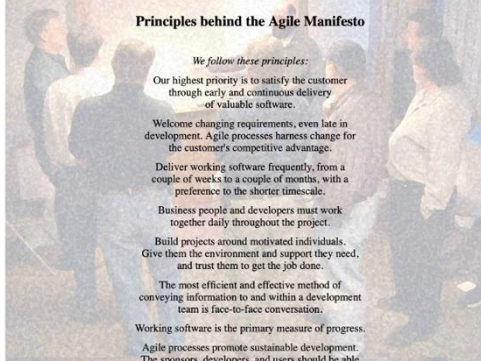
<http://agilist.guide/#19>

• References



★★★★★ 3,151

Accelerate: The Science of Lean Software a...
DevOps: Building and Scaling High Performi...
Technology Organizations
amzn.to



Principles behind the Agile Manifesto
agilemanifesto.org

The Three Ways

- The First Way: Flow/Systems Thinking
- The Second Way: Amplify Feedback Loops
- The Third Way: Culture of Continual Experimentation and Learning

Deploying Value and Recovery From Failure

- Reliability of value deployment
 - Less than 10-15%, quick fixes, rollback
- Impact of production challenges
 - Not often, rarely in prod, focus on quality
- Process to recover from failure
 - Repeatable recovery, team knows how
- Mean time to recovery
 - Prod issues solved same day

What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Reliability of value deployment	Any time the team is deploying value to customers, there is a high likelihood (40%+) that there are issues that may require a fix or some amount of rework to correct issues.	When deploying value to customers, less than a third of the time there are issues that may require a fix or some amount of rework to correct issues.	When deploying value to customers, less than 10 or 15% of the time there are issues that may require a fix or some amount of rework to correct issues.
Impact of production challenges	There are frequently issues with previously released work that it distracts the team at least once a month, causing other issues for the team.	There are issues discovered in production often enough that it occupies the team's time at least once a quarter, causing downtime of systems or rework for the team.	Issues are rarely discovered in production. The team has a thorough testing and focus on quality that eliminates most issues.
Process to recover from failure	When issues in production are found, there is no repeatable process to react and recover. In fact, there may be some degree of chaos to scramble to fix the issue.	When issues in production are found, there is a process evolving to react and recover. For example, after being notified, the team is getting better at focusing on the emergency.	When issues in production are found, there is a repeatable process to react and recover. For example, after being notified, the team drops everything to focus on the emergency. It is routine and organic, everyone on the team knows what to do.
Mean time to recovery	After issues are found in production, it can take more than a few days, a week or longer even, to respond, fix, and inform those that the issue has been resolved.	After issues are found in production, they are often handled within a day or two.	After issues are found in production, they are often handled the same day.

Deploying Value and Recovery From Failure



<http://agilist.guide/#20>

- References

Quality Embedded in Work Habits

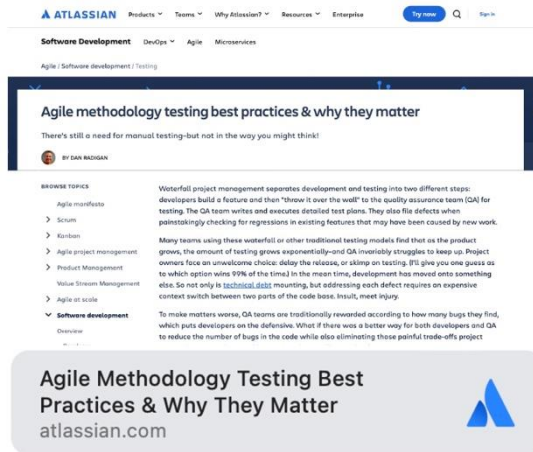
- Quality and team roles
 - No dev vs tester, part of process not roles
- Manual vs automated testing process
 - At least 80% product tests automated
- Philosophy of testing
 - Included in how they work, pairing, no bugs

What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Quality and team roles	Quality is a separate function with distinct skills on the team (or may even be off team). There are distinct handoffs between team members to complete development vs quality work.	The team considers testing part of everyone's responsibility. While there may be distinct titles and roles, the team strives to blur the line between development and quality work, using peer reviews, pairing, or skill sharing.	The team no longer distinguishes between developer and tester. It has created processes to enable quality processes to be part of how they work, not in the roles.
Manual vs automated testing process	The testing performed is almost entirely manual, not automated through repeatable processes.	There is at least 50% of the work product covered by automated tests that run regression tests over common usage patterns. Some manual testing is required to ensure greater quality.	There is at least 80% of the work product covered by automated tests that run the necessary tests. Only the more difficult to automate tests or recently completed work requires manual testing to ensure greater quality.
Philosophy of testing	Testing and quality philosophy is reactionary, not preparatory. The team has not yet embraced test driven development techniques or how to bake quality into their process.	The team considers testing and quality concerns as they estimate and as they approach any new work.	The team embraces advanced techniques to make quality part of everyone's responsibility. They are pair programming, mobbing, or otherwise tightly coupling the build and quality processes, relying on each other to validate work in near-real time and virtually eliminate production bugs.

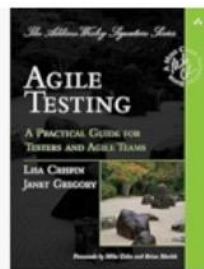
Quality Embedded in Work Habits

<http://agilist.guide/#21>



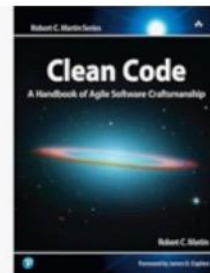
The screenshot shows the Atlassian website interface. At the top, there's a navigation bar with 'ATLASSIAN' and various menu items like 'Products', 'Teams', 'Why Atlassian?', 'Resources', and 'Enterprise'. Below that, the main content area features the article title 'Agile methodology testing best practices & why they matter' by Dan McGinn. A sub-headline reads 'There's still a need for manual testing-but not in the way you might think!'. The article text discusses the Waterfall model, the exponential growth of testing in Agile, and the importance of QA teams. A sidebar on the left lists 'BROWSE TOPICS' such as Agile manifesto, Scrum, Kanban, and Agile project management. At the bottom of the article preview, there's a call to action: 'Agile Methodology Testing Best Practices & Why They Matter' with the Atlassian logo and the URL 'atlassian.com'.

• References



★★★★★ 318

Agile Testing: A Practical Guide for Testers and Agile Teams
amzn.to



★★★★★ 6,232

Clean Code: A Handbook of Agile Software Craftsmanship
amzn.to



★★★★★ 3,058

The Pragmatic Programmer: Your Journey To Mastery, 20th Anniversary Edition (2nd Edition)
amzn.to

Work Product Organization, Promotion, and Mgt

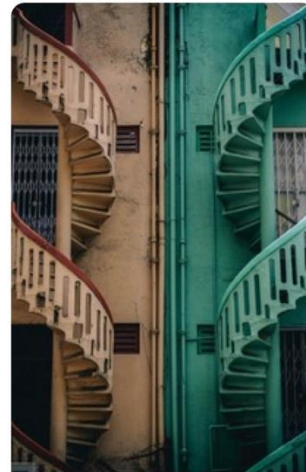
- Separated by function
 - Has all environments, reliable, sized, secure
- Process of promoting work
 - Approval process, movement, quality checks, ci/cd
- Work integration
 - Auto build process on check in, integration environ
- DevOps automation
 - All moves between environs are auto'd, kept current

What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Separation by function	The team only has individual work environments per team member and a single production environment for finished work.	The team has an environment for integrating work from their individual environments with a quality process focused on evaluating the combined work. If there are multiple quality processes, there are usually separate environments for each.	The team has all the quality and integration environments they need for their work which are reliable, stable, properly sized, and properly maintained. Additionally, they have an environment that mimics the final production environment for greater accuracy and confidence in quality.
Process of promoting work	There is no official process, rules, or procedure that the team follows for the movement of work product from one environment to another.	The team has established the guidelines or procedures for moving work from one environment to another that is reliable and deliberate.	The team has quality checks, named responsibilities, or approval processes in place to ensure a proper and high-quality movement of work between any of the environments. If those checks are not satisfied the team does not move work prematurely.
Work integration	Work is not integrated before moving completed work into a production environment. Each individual team member is responsible for that process.	Work from individual team members is coordinated and combined into a working product in a non-production environment to validate and catch mistakes before deploying to production. This process requires an individual or specific skill on the team to do the integrating.	Any team member can and often does combine the current working product from all team members into an integrated environment to do their own quality checks.
DevOps automation	The movement of work product between any environments is completely manual and subject to human error.	Portions of the process from individual team member environments to the production environment are automated in some way but require multiple manual steps. Automated processes are mostly kept up to date as things change.	The entirety of moving work product between environments is automated and kept diligently up to date as things change. There are only rare exceptions, for example when it is desired to require a deliberate approval before continuing the automation process.

Work Product Organization, Promotion, and Mgt

<http://agilist.guide/#22>

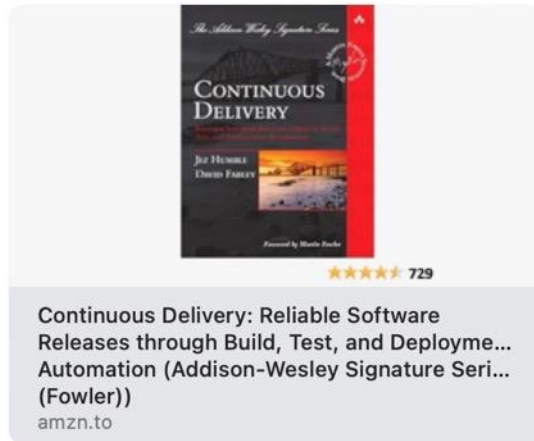


Deployment Choice: Code Promotion vs Artifact Promotion
hackernoon.com

- References



The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win 5th Anniversary Edition
amzn.to



Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment (Addison-Wesley Signature Series (Fowler))
amzn.to



The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations
amzn.to

Automated Dev Processes and Governance

- Source control usage
 - Tagging, clear commenting, per change chk/in
- Release management
 - Reliable cadence, release notes + cust lang
- Continuous integration
 - Build automation, multiple times a day + tests
- Governance
 - Auditable, builds recreated, proof of tests

What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Source control usage	There is no source control being used by the team to track the versions or history of the work product. If used at all, maybe snapshots of releases are captured, or only minimal use is involved.	The team is using source control to version the work product and use the tool for all changes to work files. They know how to rollback changes and use basic features of source control.	The team are using strategies that include some of the following: tagging of releases, branches for specific purposes, required commenting for clarity, or specific references back to requirements for each change.
Release management	There are no named, versioned, or scheduled releases. Updates to production work is sporadic or unpredictable.	Releases are deliberate and expectations with end users can be set in advance to provide some level of detail about what updates are coming to the product when. The team may not adamantly or reliably provide release schedules to end users but are able to and do so occasionally or often.	The release management process is regular and provides clear updates to end users of what changes are coming to the product on specific dates. Release notes are expressed in customer or end user language.
Continuous Integration	The integration of the team's work happens irregularly and is often a difficult manual process	There is a daily build process to catch any bugs as they emerge during development. The build process has started to be automated but may require some manual processing to complete.	There is a build process that is automated to run multiple times during the day, or even as anyone checks in code. That build process kicks off an automated suite of tests that make sure nothing major has broken since the last integration.
Governance	There are no rules or protocols for how source control is used to track the work product.	The team has committed to put clear commenting on each check-in to better identify merge conflicts and challenges as they occur.	Source control is treated as the master repository. All deliverables are auditable through source control. Builds can be recreated, exact lists of files that have changed between builds can be generated, and the team is requires a user story id for every check-in to make this process seamless.

Automated Dev Processes and Governance



What is release management?
servicenow.com



Essentials of administration and governance
with GitHub Enterprise Cloud
resources.github.com

<http://agilist.guide/#23>

ATLASSIAN

How Atlassian ensures governance and compliance

A quick guide from our executive team to yours



GEORGE TOTEV
Head of Risk and Compliance



GUY HERBERT
Risk and Compliance Manager



BILL MARRIOTT
Trust and Security Manager

As your enterprise grows, your security, governance, and compliance protocols have to grow with it. If not, you can find yourself caught in legal matters, subject to a significant data breach, or even worse, you can threaten your brand's integrity and your customers' trust.

So what's the best way to achieve top-notch enterprise security, governance, and compliance? For starters, the tools you use and the systems you have in place are critical to success. We sat down with Atlassian's security leadership team to gain insight into how we've scaled our security and compliance practices, and it boiled down to three major themes.

1. Change and release management - 2 steps to success

In the words of George Totev, Atlassian's Head of Risk and Compliance, "As the scale and complexity of your environment grows, so does the impact of incorrect changes."

How Atlassian ensures governance and compliance
atlassian.com



References



★★★★★

Embracing DevOps Release Management:
Strategies and tools to accelerate continuous
delivery and ensure quality software
deployment
amzn.to



Continuous integration best practices
about.gitlab.com

Team Process for Skill Sharing

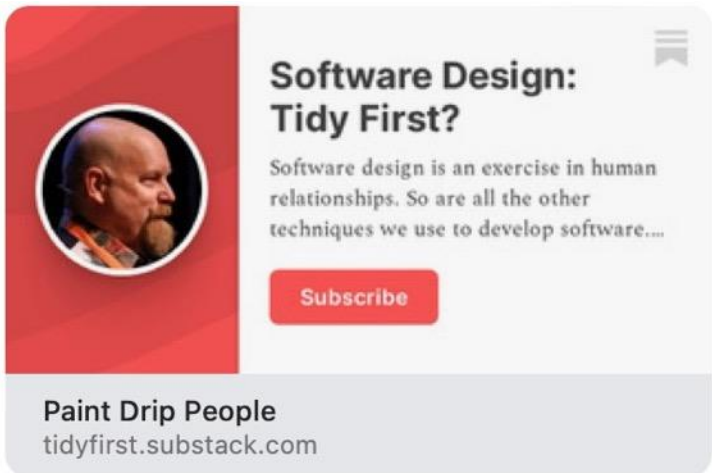
- Pairing
 - Priority to share skills, learn together
- Learning backlog
 - Part of how you work, opptys kept current
- Time and dedication
 - Space for learning, part of routine, support
- Specialists vs generalists
 - Evolved to mostly generalists, no single POF

What are the maturity levels for the behaviors on your team?

	Low	Medium	High
Pairing	The team works in silos and generally keep to themselves, not maybe even aware of the total skillsets the team has collectively.	The team has started to share skills with each other and are aware of the unique skillsets that exist in individuals they should work to share.	The team has made it a priority to share skills with each other. There is a process to either pair or mob or find time each week to learn from each other.
Learning backlog	If there are any learning plans, they are individual and often in the form of training that they and their manager have discussed.	The team has started a learning backlog that represents what the team is interested in spending their time learning.	The team keeps a learning backlog up to date that represents what the team is interested in spending their time learning and revisits it regularly.
Time and dedication	The team does not make learning a priority or feel they can even afford the time in most cases.	The team <u>makes an effort</u> to pair senior team members with newer or more junior ones to help share skills. They also sometimes work on items in their learning backlog, but it is not a consistent or regular routine.	The team has made learning and skill sharing part of their routine, dedicating time frequently and making it part of how they work.
Specialists vs. generalists	Most of the team feels they have one main <u>skill</u> and other skills are secondary or less preferred, not park of their singular core strength.	The team is developing more generalists, with a portion of the team better able to tackle any item in the backlog as opposed to only certain team members being able to do certain things.	The team is starting to look more like a team of generalists with at least half of the team able to tackle any item in the backlog.

Team Process for Skill Sharing

<http://agilist.guide/#24>



Software Design: Tidy First?

Software design is an exercise in human relationships. So are all the other techniques we use to develop software....

Subscribe

Paint Drip People
tidyfirst.substack.com



Rubber duck debugging

en.wikipedia.org

- References



Training from the Back of the Room!
65 WAYS TO STEP ASIDE AND LET THEM LEARN
SHARON L. BOWMAN
Foreword by Jeff DeGraaf

★★★★★ 228

Training From the Back of the Room!: 65 Wa...
to Step Aside and Let Them Learn
amzn.to